Tik-110.505

Methods of Cryptography:
”Interplay between Cryptography and Game Theory”

Lecture 2: A Cryptographic Solution to a Game-Theoretic Problem

Helger Lipmaa
Helsinki University of Technology
helger@tml.hut.fi

February 14, 2001

Today's literature

- [DHR00] Yevgeniy Dodis, Shai Halevi and Tal Rabin, "A Cryptographic Solution to a Game Theoretic Problem" , CRYPTO 2000.

`http://www.tml.hut.fi/~helger/teaching/crypto-gametheory/`

Game Theory Reminder

- We have a couple of selfish, rational, players, following some probabilistic strategies $s_i(\cdot)$, so as to get maximum payoffs.
- *Equilibria* = A tuple of strategies, where assuming other people are following their strategies, no participant has incentive to change strategy.
- Nash equilibria: strategies are independent. Easiest to implement!
- Correlated equilibria: strategies are not independent. Potentially better payoffs.

Correlated equilibria

- Implemented classically by using mediator, a *trusted third party*, who recommends some actions A_i to players.

- After recommendation A_{Alice}, Alice knows conditional distributions $s_{\mathsf{Alice}}(\cdot|A_{\mathsf{Alice}})$ over the actions of the other player, but nothing more.

- Since we have an equilibria and a *trusted* third party, Alice should have no incentive to deviate from the recommendation.

“Chicken” game

Alice\Bob	Chicken	Dare
Chicken	(4,4)	(1,5)
Dare	(5,1)	(0,0)

- Nash equilibria: s^1 =(Chicken,Dare) — payoff $(1, 5)$, s^2 =(Dare,Chicken) — payoff $(5, 1)$, s^3 = $\frac{1}{2}$(Chicken,Dare)+ $\frac{1}{2}$(Dare,Chicken) — *fair payoff* $(2.5, 2.5)$.

- Correlated equilibrium: $\frac{1}{3}$((Chicken,Chicken) + (Chicken,Dare) + (Dare,Chicken)), *fair payoff* $(3.5, 3.5)$.

Removing the Mediator: Payoffs

- We assume that the players are computationally bounded and can communicate prior to playing the game.

- The players get an external input (*security parameter* k); their computational capabilities are assumed to be polynomial in k.

- More precisely, the players work in probabilistic polynomial time (PPT), w.r.t. the length of their first argument 1^k.

Extended games

- First, A and B involve in a two party protocol, where they have two common inputs: 1^k and the strategy profile (s_1^*, s_2^*).
- $\Rightarrow$ the profile is also PPT-computable!
- The protocol outputs suggestions A_{Alice} and A_{Bob}, to Alice and Bob.
- This is also the only information they get to know!
- Thereafter, $A(1^k, A_{\mathsf{Alice}})$ and $B(1^k, A_{\mathsf{Bob}})$ output moves.

Reminder: Punishment for Deviation

The standard *minmax* methodology of game theory:

- The game has to go on, even if one player cheats during the or does not finish the initial zero-knowledge protocol.
- Honest player chooses a strategy that gives the least payoff to the deviator when the deviator chooses his best strategy.
- It can hurt the honest player, but since it also hurts another player, it makes everybody reluctant to not to follow the protocol.
- This works, since everybody is selfish!

Goal of Cryptographic Protocol

- To replace the mediator! What did the mediator do?
  - ⋆ Given a strategy profile (a distribution on strategies), sample it: i.e., choose a pair of actions $(A_{\mathsf{Alice}}, A_{\mathsf{Bob}})$, according to it.
  - ⋆ Output A_{Alice} to Alice only, A_{Bob} to Bob only.
- $\Rightarrow$ Protocol has to sample a random pair from the profile, given the probability distribution, and output the first coordinate of it to Alice and the second coordinate to Bob.
- Nothing about the other player's recommendation would be revealed to Alice and Bob. (A zero-knowledge/witness-hiding protocol.)

Goals, more precisely

- The strategy profile is computable in the PPT time.
- In (at least simplest of the) two-player games, the profile can be described as a relatively short list of pairs $\{(A_1, A_2)\}$, where more probable pairs are replicated.
- The strategy is to randomly choose a pair from this list.
- Formal definition of the goal: given a list $\{(a_i, b_i)\}$, pick jointly a random pair (a_i, b_i), distribute a_i to Alice and b_i to Bob.

Cryptographic tools: public key encryption

- (Probablistic) public key encryption scheme is a triple (G, E, D), where $G(1^k)$ produces the pair of $(\mathsf{pk}, \mathsf{sk})$.
- Given $(\mathsf{pk}, \mathsf{sk}) \leftarrow G(1^k)$, $D_{\mathsf{sk}}(E_{\mathsf{pk}}(m, r)) = m$, $\forall m, r$
- r is random component (nonce), necessary to achieve nice properties like semantic security.
- Also necessary for the next definition.

Blindable encryption

Add two algorithms, Blind and Combine, s.t.

- For any message m' and ciphertext $c = E_{\mathsf{pk}}(m)$, $\mathsf{Blind}_{\mathsf{pk}}(c, m')$ produces a random encryption of $m + m'$, s.t. the distribution of $\mathsf{Blind}_{\mathsf{pk}}(c, m')$ is equal to that of $E_{\mathsf{pk}}(m + m')$ (under all possible random choices).

- $\mathsf{Blind}_{\mathsf{pk}}(\mathsf{Blind}_{\mathsf{pk}}(c, m_1; r_1), m_2; r_2) = \mathsf{Blind}_{pk}(c, m_1 + m_2; \mathsf{Combine}(r_1, r_2))$

Blindable encryption: examples

Modified ElGamal: $E_{\mathsf{pk}}(m, r) = (g^m h^r, g^r)$. Here, $\mathsf{Blind}_{\mathsf{pk}}((y, x), m') = (g^{m'} y, x)$ and $\mathsf{Combine}(r_1, r_2) = r_1 + r_2$:

$$\mathsf{Blind}_{\mathsf{pk}}((g^m h^r, g^r), m') = (g^{m+m'} h^r, g^r) \ .$$

Paillier: $E_{\mathsf{pk}}(m, r) = g^m r^n \mod n^2$, here $\mathsf{Blind}_{\mathsf{pk}}(y, m') = g^{m'} y$ and $\mathsf{Combine}(r_1, r_2) = r_1 r_2$.

Okamoto-Uchiyama: $E_{\mathsf{pk}}(m, r) = g^m h^r \mod p^2 q$, here $\mathsf{Blind}_{\mathsf{pk}}(y, m') = g^{m'} y$ and $\mathsf{Combine}(r_1, r_2) = r_1 + r_2$.

⋮

Semantic security

Ciphertexts corresponding to two different messages m_0 and m_1 are indistinguishable. More precisely, for any PPT algorithm A and polynomial p, there exists a k_0, s.t.

$$\begin{aligned}\Pr[(\mathsf{pk},\mathsf{sk}) \leftarrow G(1^k), (m_0, m_1) \leftarrow A(1^k, \mathsf{pk}), b \leftarrow \{0,1\} : \\ A(1^k, \mathsf{pk}, E_{\mathsf{pk}}(m_b)) = b] < 1/p(k)\end{aligned}$$

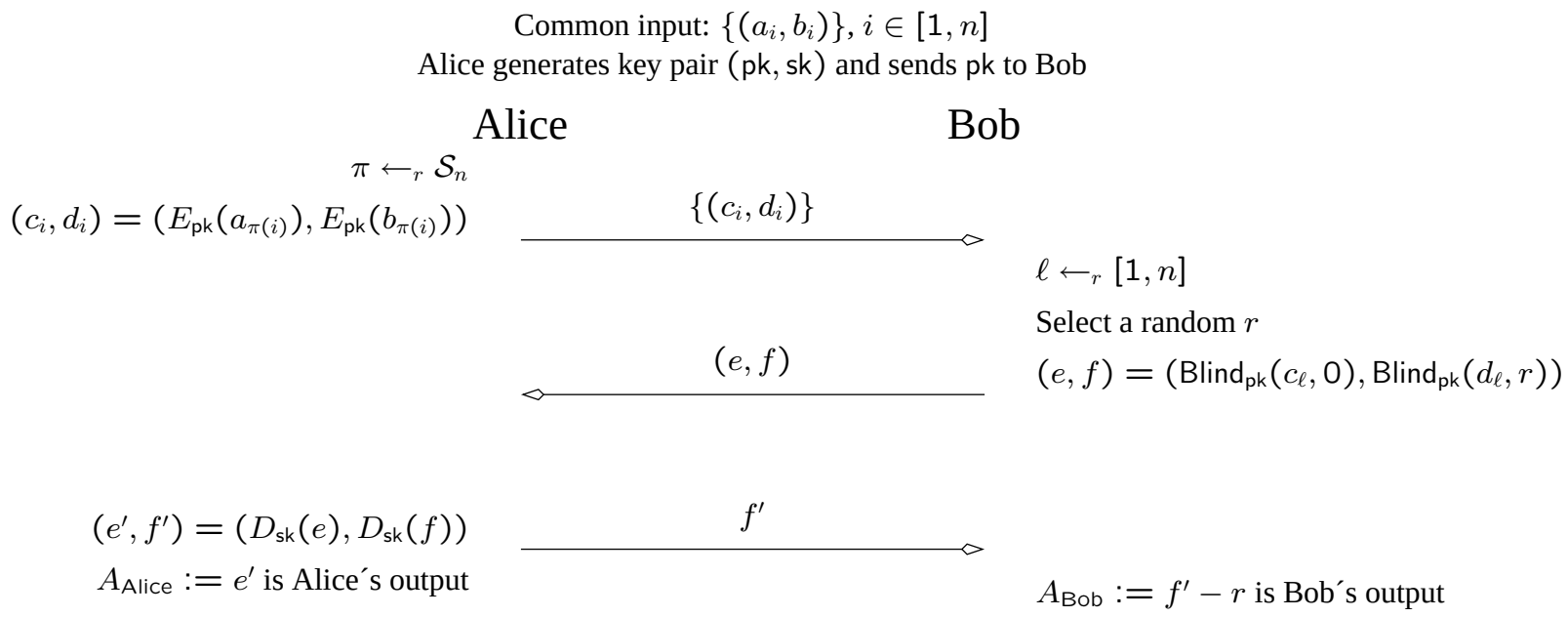
whenever $k > k_0$.

More generally, it means that given a ciphertext, A cannot extract any useful information about the plaintext.

In what follows we have a semantically secure blindable encryption scheme.

Easy protocol: players are honest-but-curious

A and B have as common input the list $L = \{(a_i, b_i)\}$. Alice generates a key pair $(\mathsf{pk}, \mathsf{sk})$ and sends pk to Bob.



Honest-but-curious: analysis

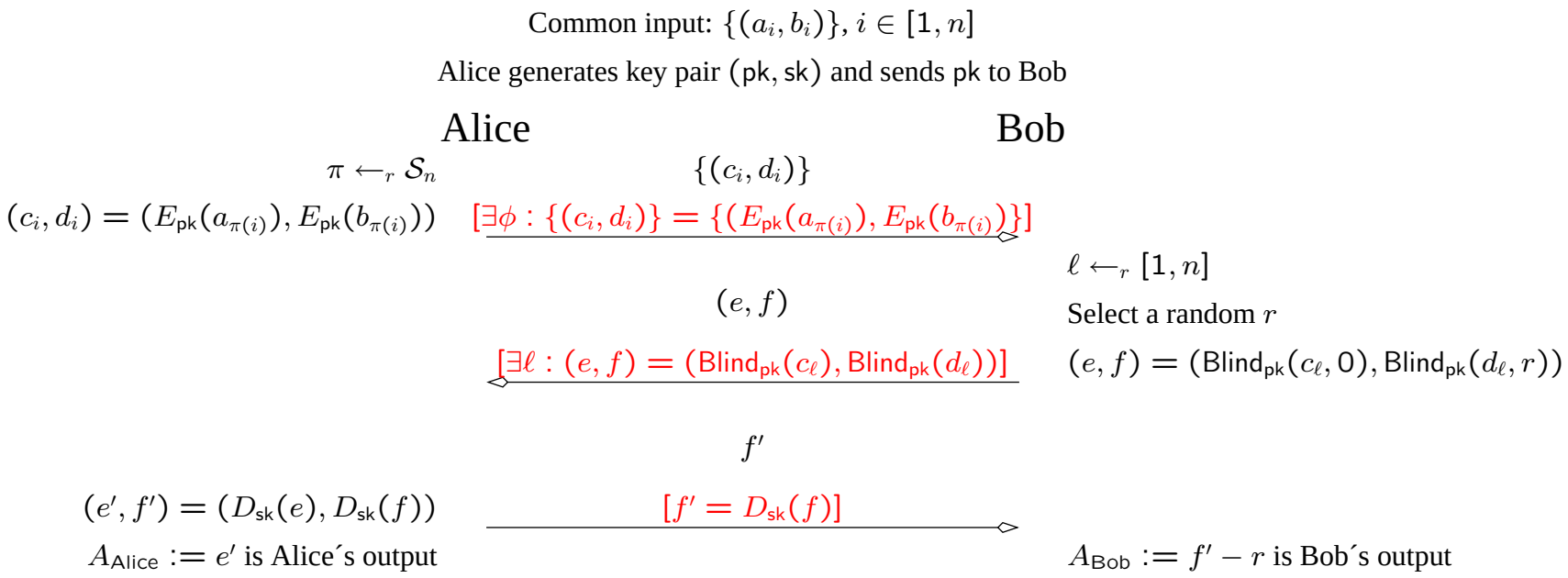
- Since the scheme is semantically secure, Bob obtains no information on the used permutation and hence on A_{Alice} except that which follows from knowledge of A_{Bob}.

- Since r is random, Alice obtains no information on A_{Bob} except that which follows from knowledge of A_{Alice}.

- Bob has incentive to pick ℓ randomly (otherwise Alice would be able to trick him), hence also the distribution is correct.

Dishonest players?

Add zero-knowledge proofs that every step is performed correctly!



Proof of proper decryption

Alice has to prove, given (y, m), that $D_{\mathsf{sk}}(y) = m$.

- Proof does not have to be zero-knowledge!
- Alice does *not* know r (otherwise she could just reveal it).
- Alice sends $\{b_{\pi(i)}, r_{\pi(i)}\}$, s.t. $\{d_{\pi(i)}\} = \{E_{\mathsf{pk}}(b_{\pi(i)}, r_{\pi(i)})\}$.
- Bob verifies that $d_\ell = E_{\mathsf{pk}}(b_\ell, r_\ell)$.

Bob will get to know $\{b_{\pi(i)}\}$, but this does not give him any information on $a_{\pi(i)}$ that he does not have by knowing $b_{\pi(i)}$ alone!

Encrypted list correspondence

Alice and Bob know two lists $\{a_i\}$ and $\{c_i\}$. Alice has to prove that she knows a permutation π and nonces $\{r_i\}$, s.t. $c_i = \mathsf{Blind}_{\mathsf{pk}}(a_{\pi(i)}, 0; r_i)$.

Intuition behind the proof (Fig. 3 in the paper):

- Alice generates random permutation ρ and random nonces $\{s_i\}$, and depending on Bob's challenge $\in \{0, 1\}$, reveals either $(\rho, \{s_i\})$ or $(\pi \circ \rho, \{\mathsf{Combine}(r_{\rho(i)}, s_i)\})$.

- Since ρ and $\{s_i\}$ are random, Bob gains no knowledge of π or $\{r_i\}$.

- Alice's best strategy in cheating is to guess Bob's challenge (probability $1/2$.

$\Rightarrow$ constant-error three-round ZK proof. All such protocols can be transformed to constant-round negligible-error ZKP's.

Proof that (c', d') is correct

Bob needs to prove that (c', d') is a blinded version of some (c_ℓ, d_ℓ) for some ℓ, without revealing ℓ.

He can use the previous protocol, by first picking a random permutation τ and then letting $\ell = \tau(1)$.

Conclusions

More details in the paper.

Do you want to sometimes go through the general constructions of negligible-error ZKP's?

Next few times: auctions + game theory.