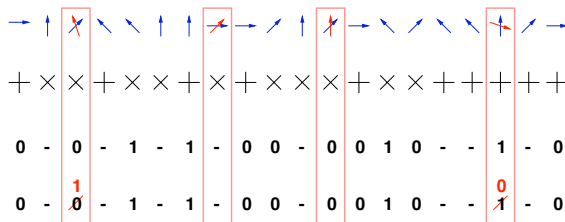


Quantum Cryptography: A Glimpse

BICI-INDAM 2005 International PhD School on Mathematical Aspects of Modern Cryptography, Sept. 4–9, 2005, Bertinoro.

Quantum cryptography: example



first row: photons sent by Alice
 second row: bases selected by Bob
 third row: bits generated by Alice
 fourth row: bits generated by Bob

Quantum cryptography: some explanations

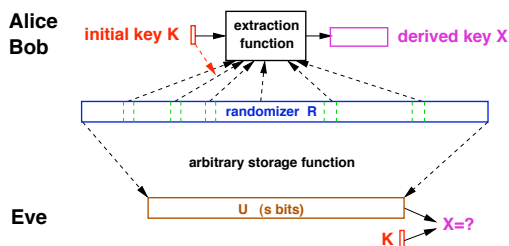
- Alice and Bob are connected by a conventional insecure but authenticated communication channel as well as an optical fiber allowing Alice to send photons to Bob. Eve has access to the fiber.
- The polarisation of a photon can encode information, but due to the laws of quantum physics, only two states can reliably be distinguished by any measurement. Hence one can transmit reliably only 1 bit of information by encoding the two bits in orthogonal polarisations.
- Two different bases for sending a bit are defined: the horizontal/vertical basis and the diagonal (45°/135°) basis.
- Alice sends a sequence of random bits, each in a random basis. Eve cannot measure exactly which of the 4 states was transmitted.
- Bob measures each received photon in random basis and tells Alice which bases he has used. Alice announces for which bits Bob used the right basis and hence knows Alice's bits. Using error correction and privacy amplification, Alice and Bob can extract a secret key.
- One can prove that Eve has only a choice between performing too strong measurements and therefore being detected by Alice and Bob with high probability, or obtaining essentially no information about the derived key.

The Bounded-Storage Model

BICI-INDAM 2005 International PhD School on Mathematical Aspects of Modern Cryptography, Sept. 4–9, 2005, Bertinoro.

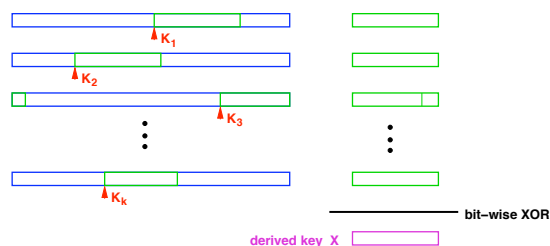
The bounded-storage model (BSM)

Goal: Expansion of short **initial key K** into a long **derived key X**.



Question: What about quantum storage?

A scheme for the bounded-storage model



Theorem [DM02]: For all storage functions and every $\mu < 1$, $d(X|UK)$ is negligible if $s < \mu \cdot H_\infty(R)$ (for sufficiently long randomizers).

The hybrid BSM

Goal: **Protection against future advances in computing** (e.g. quantum computers, better algorithms)

Definition: A key-agreement protocol is **computationally secure** (against passive eavesdropping) if, when given the transcript, it is computationally infeasible to distinguish K from an independent random key.

Proposition: If a computationally secure key-agreement scheme (e.g. Diffie-Hellman) is used to generate K , then the BSM-scheme remains secure even if Eve later obtains infinite computing power.

Proof (sketch): Since the scheme is secure even if Eve learns K

The hybrid BSM (2)

Theorem [DM04]: There exists a computationally secure key-agreement protocol for which the BSM is insecure.

Proof (sketch):

A secure KA scheme is enhanced artificially by including in the transcript queries of a computationally secure PIR scheme, where the database (of PIR) corresponds to the randomizer R .

The queries allow to access the relevant (for K) randomizer bits, without revealing which bits are accessed.

The PIR replies are stored by Eve.

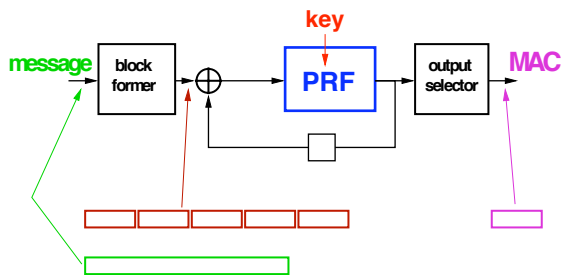
Indistinguishability-based Security Definitions and Proofs

BICI-INDAM 2005 International PhD School on Mathematical Aspects of Modern Cryptography, Sept. 4–9, 2005, Bertinoro.

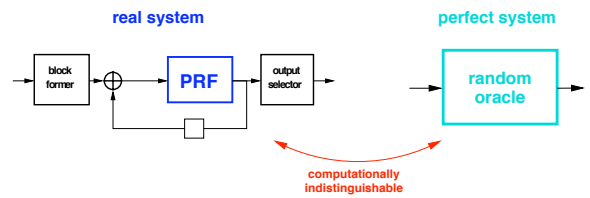
Outline

- A motivating example: CBC-MAC
- Indistinguishability
- Random systems: modeling input/output behavior
- Monotone conditions in a random system
- Conditional equivalence of random systems
- Indistinguishability proofs
- Adaptive vs. non-adaptive strategies
- Some applications: CBC-MAC, generalized Luby-Rackoff
- Quasi-randomness

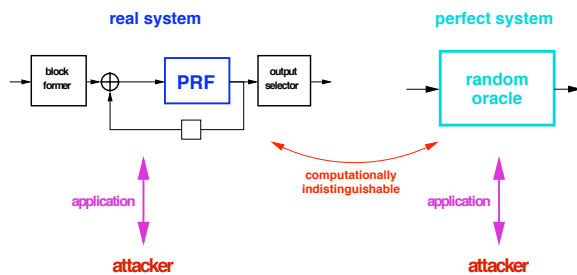
Motivating example: CBC-MAC



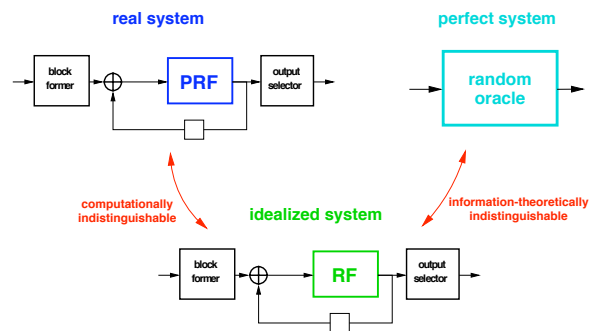
Goal of security proof for CBC-MAC [BKR94, ...]



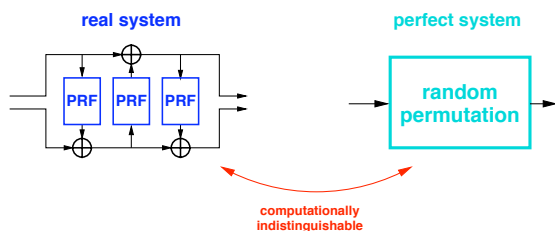
Goal of security proof for CBC-MAC [BKR94, ...]



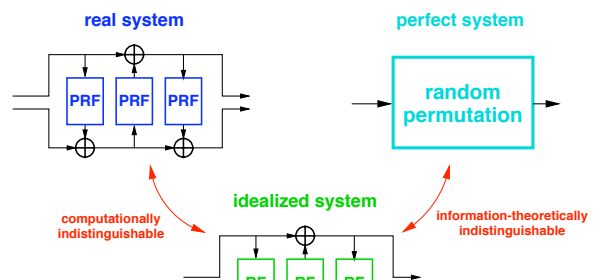
Goal of security proof for CBC-MAC [BKR94, ...]



Example: Luby-Rackoff pseudo-random permutations



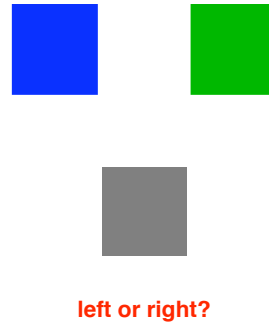
Example: Luby-Rackoff pseudo-random permutations



Goals of framework:

- General framework for indistinguishability proofs
- Abstraction \Rightarrow generality and simplicity
- Unify, simplify, generalize, strengthen known results
- New generic results on indistinguishability
- Quasi-random systems: The general design problem

Distinguishing two objects:



Distinguishing two types of numbers

Set A:

2048-bit integers with exactly **2** prime factors, each with at least 512 bits.

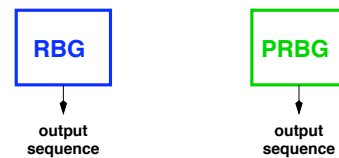
Set B:

2048-bit integers with exactly **3** prime factors, each with at least 512 bits.

374095762974511873398056743981753957783254673845967825364509871
 365295584882333644985766091852825640501638759879538762635485678
 243091425765253648526374099125231764748985576600963327393947586
 123498750533495862054987746524351089758393218367443278968764534
 3127364987564354675092736565475849823142537584950243685261

left or right?

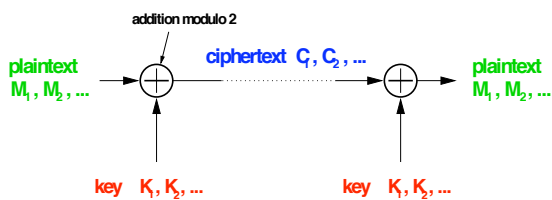
Random vs. pseudo-random bit generator



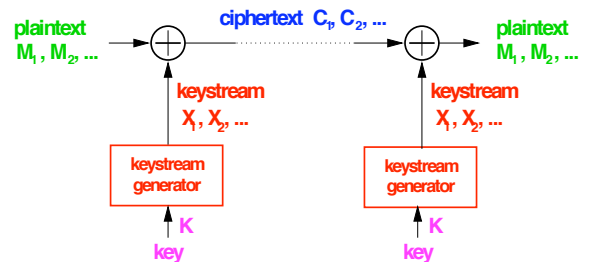
10110001110111100100111010001000011101100101110010111010001101
 000011011010111101010001101011010100100101011110101000001101101
 111000111011000101111010010101101001010110000101011010101101001
 110011001001100010110100011100101010001011010100001111000101010

left or right?

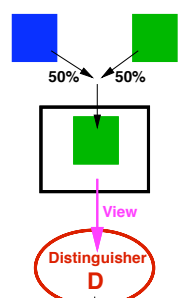
One-time pad



Binary additive stream cipher



Distinguisher's advantage: two equivalent definitions

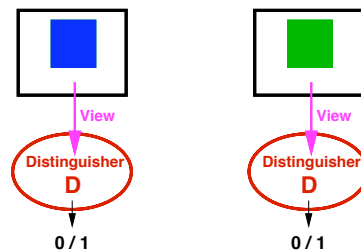


One random experiment with a 50-50 selection.

D's task: Guess left/right.

$\text{Prob}(\text{correct guess}) = 0.5 + \alpha/2$

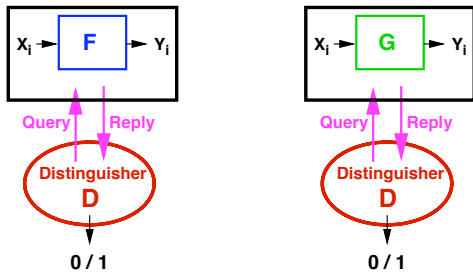
$\alpha = \text{advantage of } D$



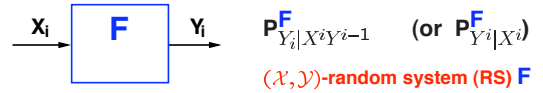
Two separate random experiments.

D outputs 0 or 1.

Distinguishing two systems F and G:



$$\text{Advantage} = |\text{Prob}^F(1) - \text{Prob}^G(1)|$$



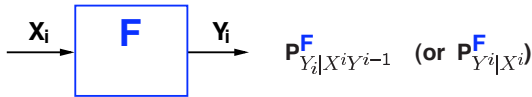
Deterministic, stateless: Function $\mathcal{X} \rightarrow \mathcal{Y}$

Probabilistic, stateless: $(\mathcal{X}, \mathcal{Y})$ -random function **F**:
Random variable taking as values functions $\mathcal{X} \rightarrow \mathcal{Y}$.

Deterministic, stateful: $(\mathcal{X}, \mathcal{Y})$ -automaton **F**:
Seq. f_1, f_2, \dots with $f_i: \mathcal{X}^i \rightarrow \mathcal{Y}$ and $Y_i = f_i(X_1, \dots, X_i)$.

Probabilistic, stateful: $(\mathcal{X}, \mathcal{Y})$ -random automaton (RA) **F**:
Sequence F_1, F_2, \dots of random variables.

Examples of random systems

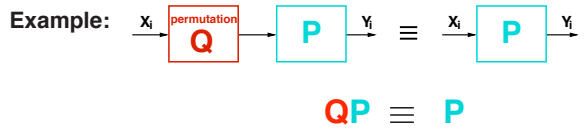


- Uniform random function R
- Uniform random permutation P
- Beacon B (ignores input, special case of a source).
- CBC-MAC
- Random oracle O (perfect MAC)

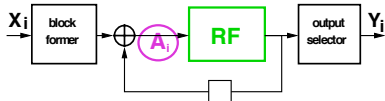
Equivalence of random automata



$$F \equiv G \iff P_{Y_i|X^i Y^{i-1}}^F = P_{Y_i|X^i Y^{i-1}}^G \text{ for } i \geq 1.$$



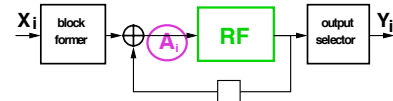
Random systems with monotone internal conditions



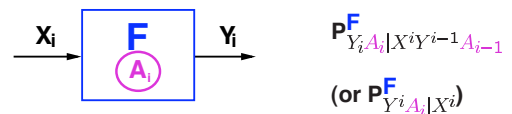
A_i = all inputs to RF are distinct, up to message X_i (except trivial cases).



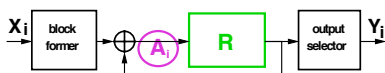
Random systems with monotone internal conditions



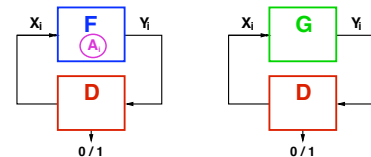
Modeling random systems with monotone events:



Conditional equivalence



$$\text{CBC(R)} | A \equiv O$$



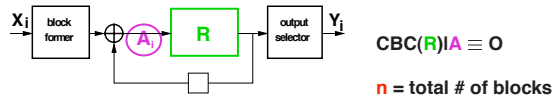
Optimal advantage: $\Delta_k(F, G) := \max_D |P^{DF}(1) - P^{DG}(1)|$

Maximal prob. of provoking \bar{A}_k : $\nu(F, \bar{A}_k) := \max_D P^{DF}(\bar{A}_k)$

same, but non-adaptive: $\mu(F, \bar{A}_k) := \max_{x^k} P_{\bar{A}_k}^F(x^k)$

Theorem: $F|A \equiv G \implies \Delta_k(F, G) < \nu(F, \bar{A}_k)$

Security proof for CBC-MAC



$\overline{A_k}$ = all inputs to RF are distinct, up to message X_k

$$\Delta_k(\text{CBC}(\mathbf{R}), \mathbf{O}) \leq \nu(\text{CBC}(\mathbf{R}), \overline{A_k}) = \mu(\text{CBC}(\mathbf{R}), \overline{A_k})$$

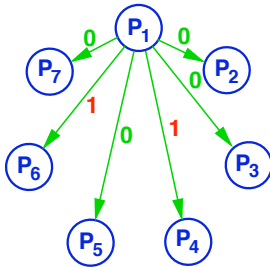
Lemma: Block former prefix-free $\Rightarrow \mu(\text{CBC}(\mathbf{R}), \overline{A_k}) \leq \frac{1}{2}n^2 2^{-l}$.

Theorem: $\Delta_k(\mathbf{F}, \mathbf{R}) \leq d(k) \Rightarrow \Delta_k(\text{CBC}(\mathbf{F}), \mathbf{O}) \leq \frac{1}{2}n^2 2^{-l} + d(n)$.

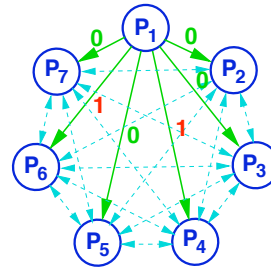
Secure Multi-Party Computation

BICI-INDAM 2005 International PhD School on Mathematical Aspects of Modern Cryptography, Sept. 4–9, 2005, Bertinoro.

Broadcast / Byzantine agreement



Broadcast / Byzantine agreement



Theorem [LSP80]: Among n players, broadcast is achievable if and only if $t < n/3$ players are corrupted.

Outlook

Starting point: LSP broadcast theorem ($t < n/3$)

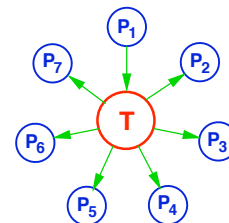
Generalization 1: Secure multi-party computation

Generalization 2: General adversary structures

(Generalization 3: General assumed primitives)

(Generalization 4: Consistency specifications)

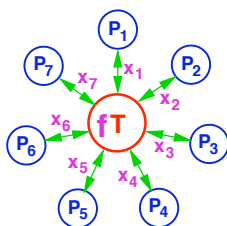
Broadcast / Byzantine agreement



Theorem [LSP80]: Among n players, broadcast is achievable if and only if $t < n/3$ players are corrupted.

Simulation of a **trusted party T**.

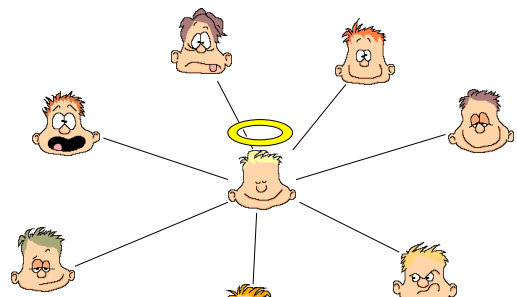
Generalization 1: Secure computation



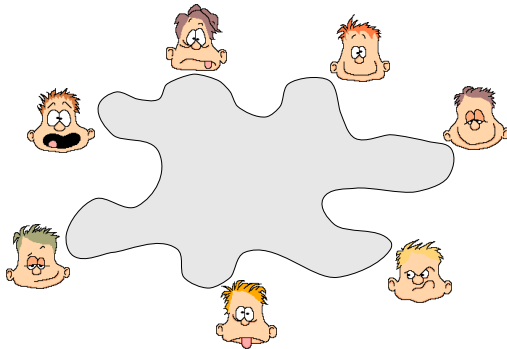
T computes a function $f(x_1, \dots, x_7)$ of the inputs.

New operations of **T**: receive secret input

Ideal solution: Involve a trusted party



Real solution: Simulation of trusted party



Some applications

- The millionaires' problem
- Preventing software piracy
- On-line auctions
- E-voting
- Secure aggregation of databases

Secure MPC: Summary of known results

Adversary types:

- **passive**: plays correctly, but analyses transcript.
- **active**: cheats arbitrarily.

Types of security:

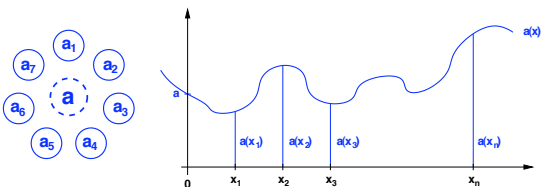
- **computational**: intractability assumptions
- **information-theoretic**: ∞ computing power

type of security	adv. type	condition
computational	passive	$t < n$
computational	active	$t < n/2$
information-theoretic	passive	$t < n/2$
information-theoretic	active	$t < n/3$

Secret sharing

(t,n)-secret sharing: Share a value **a** among n players such that

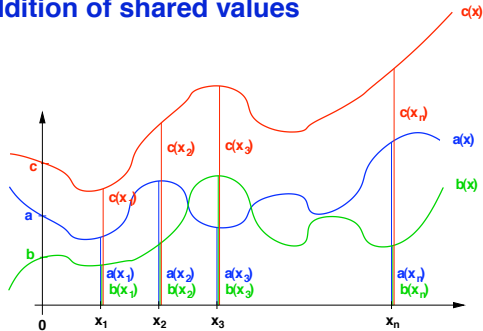
- any **t** players have no information about **a**,
- any **t+1** players can reconstruct **a**.



Every player is assigned a fixed value x_i from a finite field.

To share the value **a**, choose a random polynomial **a(x)** of degree **t** such that **a(0)=a**. The share of the *i*-th player is **a(x_i)**.

Addition of shared values



$$c(x_i) = a(x_i) + b(x_i) \Rightarrow c(x) = a(x) + b(x) \Rightarrow c(0) = a(0) + b(0)$$

From linear to general computations

A **linear functions** **f** on several shared values can be computed by **locally** computing **f** on the shares.

Multiplication:

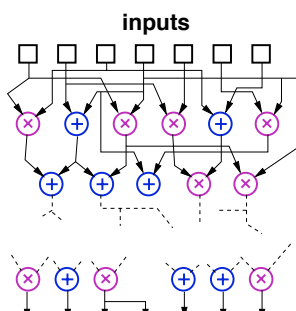
Locally multiplying shares ($c(x_i) = a(x_i)b(x_i)$) fails because the degree of the polynomial $c(x) = a(x)b(x)$ is **2t** instead of **t**.

However, **c(0)** can be interpolated by a **linear function** from the share products $a(x_i)b(x_i)$:

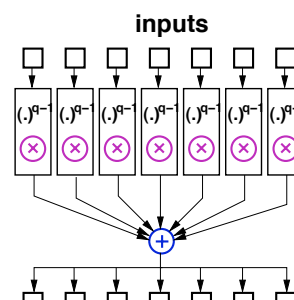
$$c(0) = \sum_{i=1}^n l_i a(x_i)b(x_i) \quad \text{for some } l_i \quad (*)$$

- Each player P_i shares $a(x_i)b(x_i)$
- A sharing of **c(0)** is obtained by **locally** computing (*).

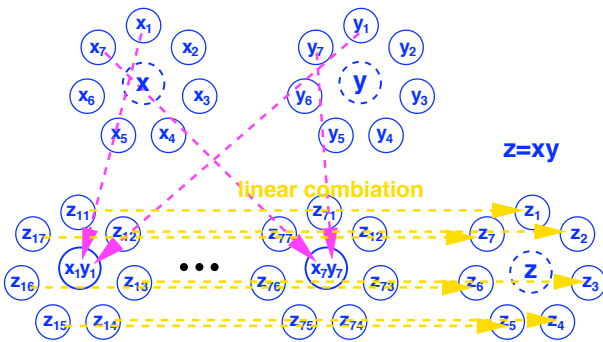
Computation as an arithmetic circuit



The circuit for adding votes (e-voting)

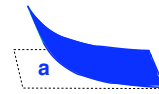
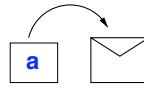


Multiplication of shared values x and y



Commitment schemes

Mechanical analogs:



Notation:

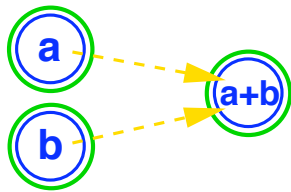


color indicates the committed player

A **commitment scheme** allows a player to **commit** to a value a , such that he can later **open** the commitment, and

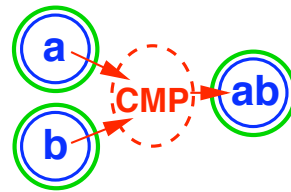
- **Binding property:** The player cannot open a committed value in two different ways.
- **Hiding property:** The other players obtain no information about the committed value from the commitment.

Operations on Commitments



Exploit **homomorphic property** of commitment scheme.

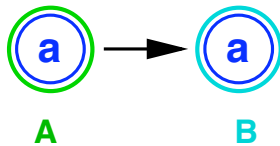
Operations on Commitments



Commitment Multiplication Protocol (**CMP**):

Interactive protocol involving a **zero-knowledge proof**.

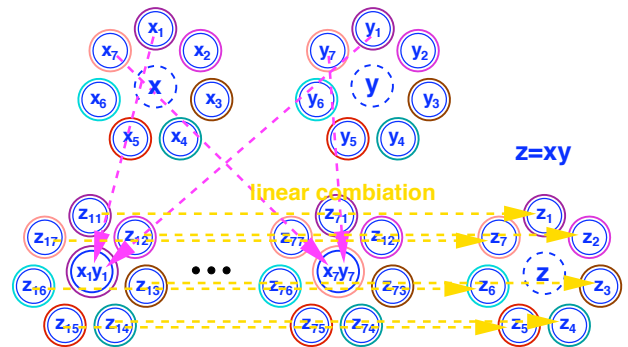
Operations on Commitments



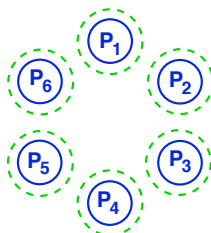
Commitment Transfer Protocol (CTP):

Before: Player **A** is committed to a .

After: Player **B** is committed to a .



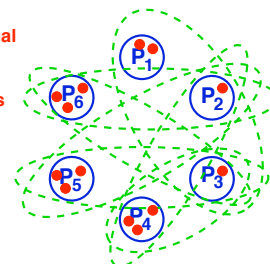
Generaliz. 2: General adversary structures



Adversary structure:

Generaliz. 2: General adversary structures

$n=13$ virtual players, up to 4 cheaters



Adv. structure: $\Delta = \{123, 34, 25, 26, 35, 36, 24\}$

Generaliz. 2: General adversary structures

Theorem [Hirt-M97]: Adv. structure Δ can be tolerated iff $\forall S_1, S_2, S_3 \in \Delta : S_1 \cup S_2 \cup S_3 \neq \mathcal{P}$.

This result carries over to broadcast.

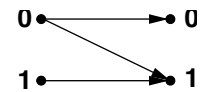
But: The efficiency is **polynomial in the size of the description of the adversary structure** (e.g. exponential for threshold structure).

Theorem [Fitzi-M98]: For any adversary structure Δ allowing broadcast, there exists a broadcast protocol with efficiency **polynomial in the number n of players**.

Generalization 3: Assumed primitives

Assumed so far: **authenticated bilateral channels**.

Weaker primitive: z-channel

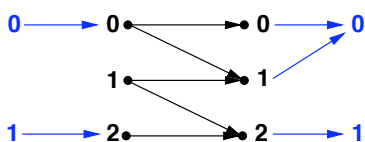


Cannot be used to obtain a reliable 1-bit channel!

Generalization 3: Assumed primitives

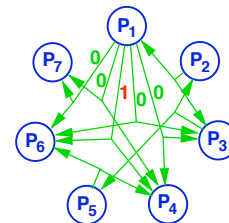
Assumed so far: **authenticated bilateral channels**.

Weaker primitive: zz-channel



Can be used to obtain a reliable 1-bit channel.

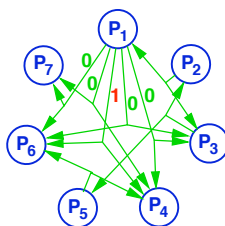
Generalization 3: Assumed primitives



Stronger primitive: BC among any **b=3** players.

Theorem [Fitzi-M00]: If BC channels for any **3** players are available, then broadcast is achievable if and only if $t < n/2$ (instead of $t < n/3$).

Generalization 3: Assumed primitives



Stronger primitive: BC among any **b=3** players.

Theorem [CFFLMM04]: If BC channels for any **b** players are available, then broadcast is achievable if and only if $t < (1 - 2/(b+1)) n$. (E.g. $b=4 \Rightarrow t < 3/5 n$)