ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Optimistic Fair Exchange Based on Publicly Verifiable Secret Sharing

Gildas AVOINE & Serge VAUDENAY

EPFL

The 9th Australasian Conference on Information Security and Privacy

13-15 July 2004, Sydney, Australia

Outline

Some Recalls on the Fair Exchange Problem

A New Fair Exchange Protocol Based on Secret Sharing

Analysis of the Protocol

Some Recalls on the Fair Exchange Problem

Two-party fair exchange: definitions

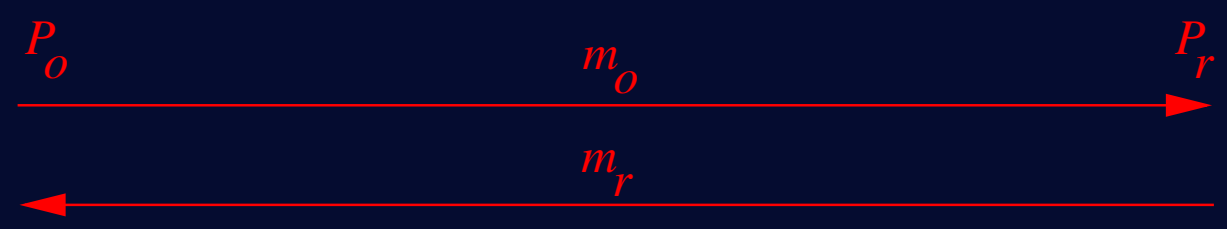
An exchange protocol between an originator P_o and a recipient P_r is a protocol in which P_o and P_r own some items m_o and m_r respectively and aim at exchanging them.

When at least one of the two participants follows the protocol, the exchange protocol ensures fairness if the exchange terminates so that either P_o gets m_r and P_r gets m_o, or P_o gets no information about m_r and P_r gets no information about m_o.

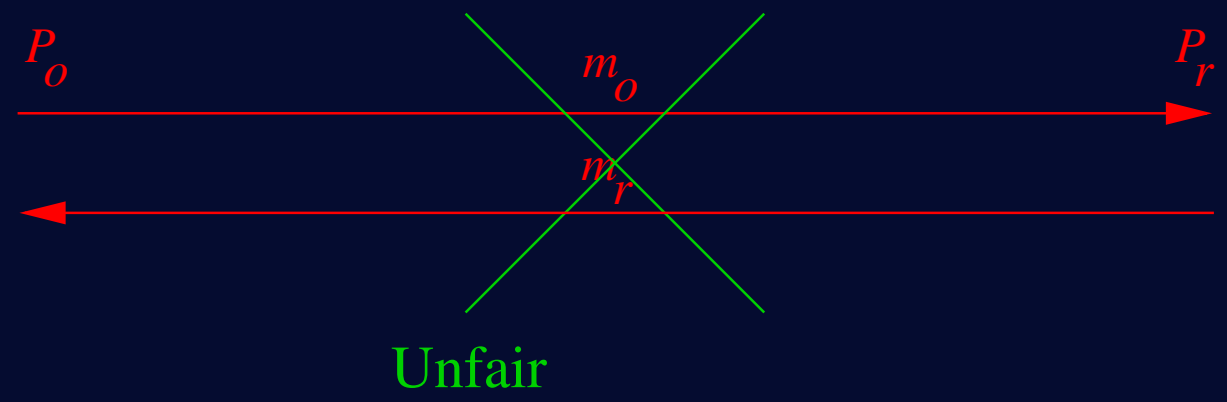
The exchange protocol ensures privacy if no other party gets any information about m_o or m_r.

The simplest exchange protocol

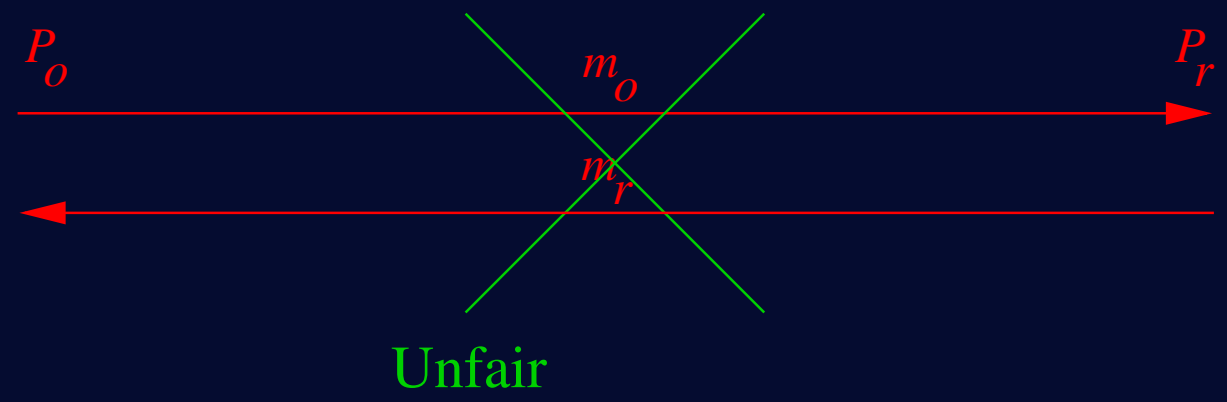
The simplest exchange protocol



The simplest exchange protocol



The simplest exchange protocol



[Even and Yacobi, 1980]

Fair exchange: classification

- FE with an on-line Trusted Third Party (TTP)
- FE with an off-line TTP
- Gradual FE
- Probabilistic FE

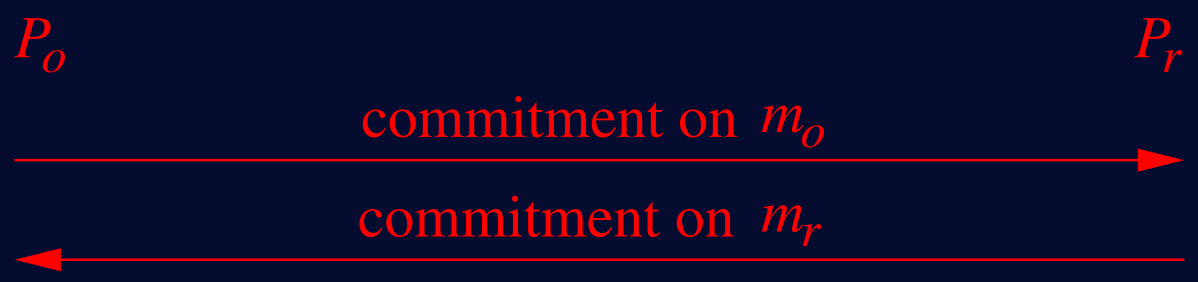
Fair exchange: classification

- FE with an on-line Trusted Third Party (TTP)
- FE with an off-line TTP
- Gradual FE
- Probabilistic FE

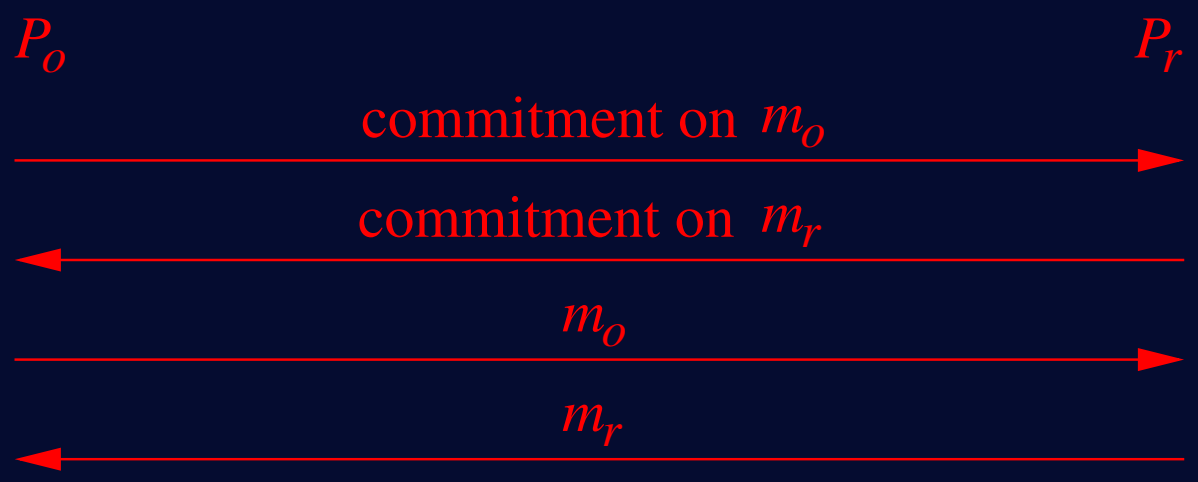
FE with an off-line TTP (main protocol)

P_o P_r

FE with an off-line TTP (main protocol)



FE with an off-line TTP (main protocol)



FE with an off-line TTP (recovery protocol)

P_o TTP P_r

FE with an off-line TTP (recovery protocol)

P_o TTP P_r

commitments on m_r and m_o $\longrightarrow$

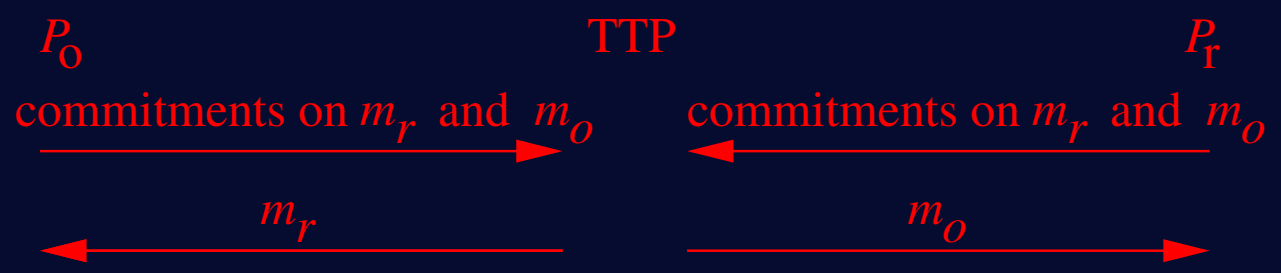
FE with an off-line TTP (recovery protocol)

P_o TTP P_r

commitments on m_r and m_o $\longrightarrow$

$\longleftarrow$ commitments on m_r and m_o

FE with an off-line TTP (recovery protocol)



A New Fair Exchange Protocol Based on Secret Sharing

Secret sharing

Secret sharing: a secret m is shared among several participants such that only some specific subsets of participants can recover m by collusion.

Verifiable secret sharing: each participant can check his own share.

Publicly verifiable secret sharing: anybody can check the shares.

PVSS: distribution stage

- The dealer generates the shares m_i of m.
- He publishes the encrypted values $E_i(m_i)$ such that only the participant P_i is able to decrypt $E_i(m_i)$.
- He publishes an information Δ which allows to prove that the distributed shares are correct i.e. they allow to recover some m verifying Δ.

$$\begin{array}{rcl} m & \rightarrow & \\ P_1, ..., P_n & \rightarrow & \end{array} \boxed{\text{Share}} \begin{array}{rl} \rightarrow & E_1(m_1), ..., E_n(m_n) \\ \rightarrow & \Delta \end{array}$$

PVSS: verification stage

- Given the P_is' public keys, the $E_i(m_i)$s, and Δ, anybody can verify that the shares allow to recover some m verifying Δ.

$$\begin{array}{rcl} E_1(m_1), ..., E_n(m_n) & \rightarrow & \\ P_1, ..., P_n & \rightarrow & \boxed{\text{Verify}} \quad \rightarrow \quad \text{true or false} \\ \Delta & \rightarrow & \end{array}$$

PVSS: reconstruction stage

- The participants decrypt their share m_i from $E_i(m_i)$.
- They pool them in order to recover m.

$$\begin{array}{r} m_{i_1}, ..., m_{i_k} \quad \rightarrow \\ P_{i_1}, ..., P_{i_k} \quad \rightarrow \end{array} \boxed{\text{Recover}} \quad \rightarrow \quad m$$

Towards a new approach

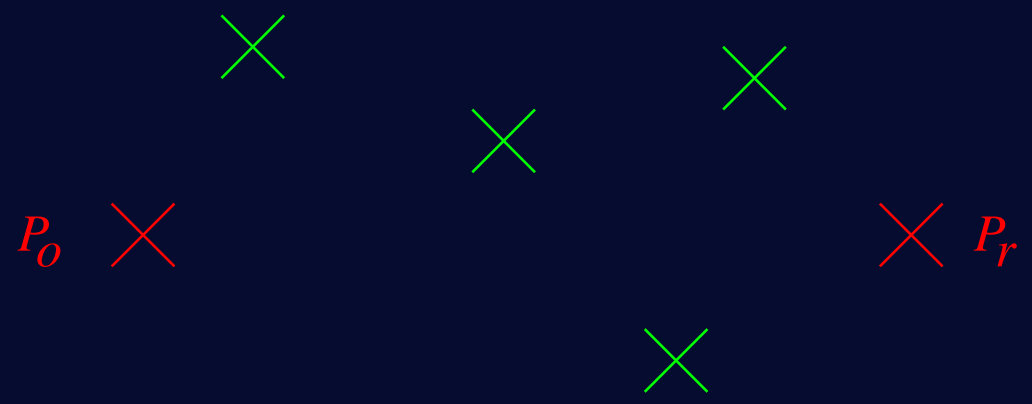
P_o $\times$ $\times$ P_r

Towards a new approach

Towards a new approach

Towards a new approach

Towards a new approach



Towards a new approach

Towards a new approach

Can we take advantage of these neighbors?

Honesty

[Avoine et al., 2004] proved that multi-party fair exchange without any form of honesty is impossible (Generalization of Even and Yacobi's results).

We assume that some participants are honest (i.e. they follow the protocol), but we don't know who they are.

Even if a participant is honest, he can be curious.

Our fair exchange protocol

Our protocol is an optimistic FE protocol: the main protocol consists of a commitment step based on a PVSS of parameter k and an exchange step.

Among the passive participants, the following behaviors exist:

- $\mathcal{B}_1$: participants who honestly collaborate with both P_o and P_r.
- $\mathcal{B}_2$: participants who may harm P_o by colluding with P_r.
- $\mathcal{B}_3$: participants who may harm P_r by colluding with P_o.
- $\mathcal{B}_4$: participants who do not collaborate at all.

Assumptions

$|\mathcal{B}_1| > 0$; and P_o and P_r know some constant k such that $|\mathcal{B}_2| < k \leq |\mathcal{B}_2| + |\mathcal{B}_1|$.

Example 1: If P_o and P_r know that there is a majority of honest participants in the network i.e. $|\mathcal{B}_1| > \frac{n}{2}$ then we take $k = \lceil \frac{n}{2} \rceil$.

Example 2: If P_o knows that at least 40% of the network is honest with him (i.e. $|\mathcal{B}_1| + |\mathcal{B}_3| \geq \frac{2n}{5}$) and P_r knows that at least 70% of the network is honest with him (i.e. $|\mathcal{B}_1| + |\mathcal{B}_2| \geq \frac{7n}{10}$) then we can take k such that $\lfloor \frac{6n}{10} \rfloor < k \leq \lceil \frac{7n}{10} \rceil$. For instance, if $n = 100$, k is chosen such that $60 < k \leq 70$.

Assumptions (Cont)

We assume that P_r knows a constant $T_{\text{max}} < +\infty$ such that messages from P_r to any participant are always delivered within a time delay less than T_{max}. We assume that all messages from honest participants are eventually delivered.

Some additional primitives

We suppose that P_o and P_r agree on the mathematical description of the items they want to exchange (e.g. descr$(m) = g^m$).

$$\begin{array}{r} m \rightarrow \\ \text{descr}(m) \rightarrow \end{array} \boxed{\text{Check}} \rightarrow \text{true or false}$$

They establish the contract:

$$\Omega = S_o(P_o \| P_r \| \text{descr}(m_o) \| \text{descr}(m_r) \| T \| k).$$

Some additional primitives

$E_o(m) \rightarrow$

descr$(m) \rightarrow$

$\boxed{\text{CheckEnc}} \rightarrow$ true or false

Some additional primitives

$E_o(m) \rightarrow$
$\Delta' \rightarrow$
descr$(m) \rightarrow$

$\boxed{\text{CheckEnc}} \rightarrow$ true or false

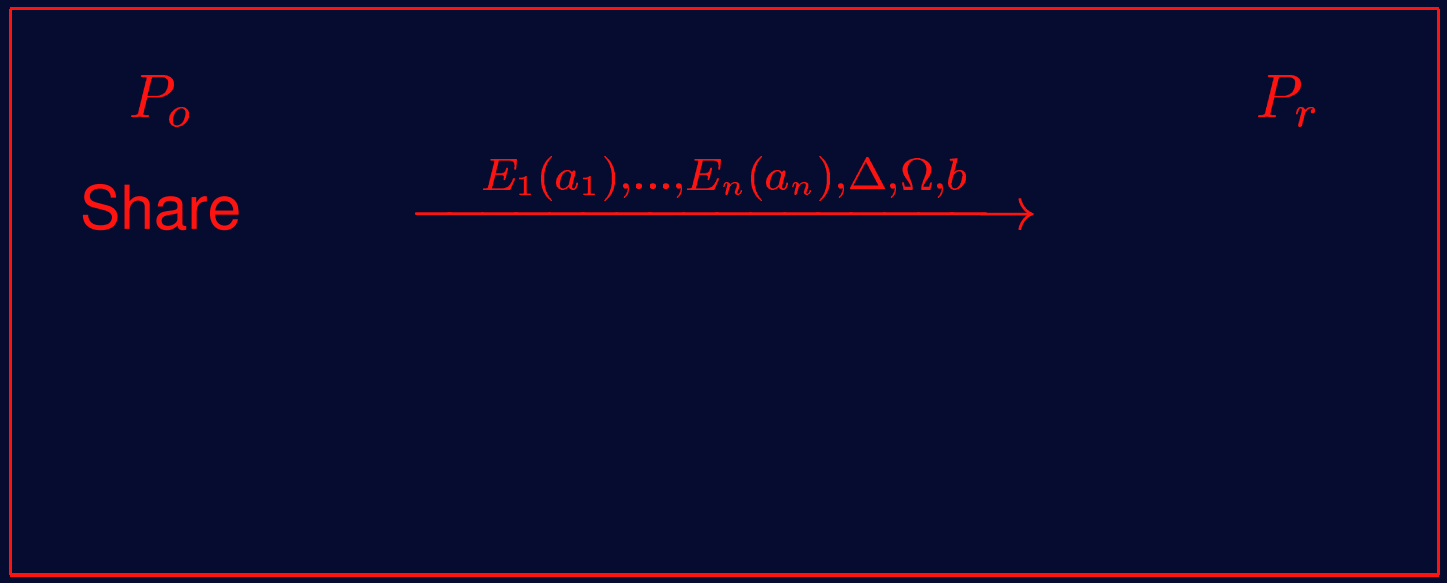
$m \rightarrow \boxed{\text{Enc}} \rightarrow E_o(m)$
$\rightarrow \Delta'$

$E_o(m) \rightarrow \boxed{\text{Dec}} \rightarrow m$

Recap

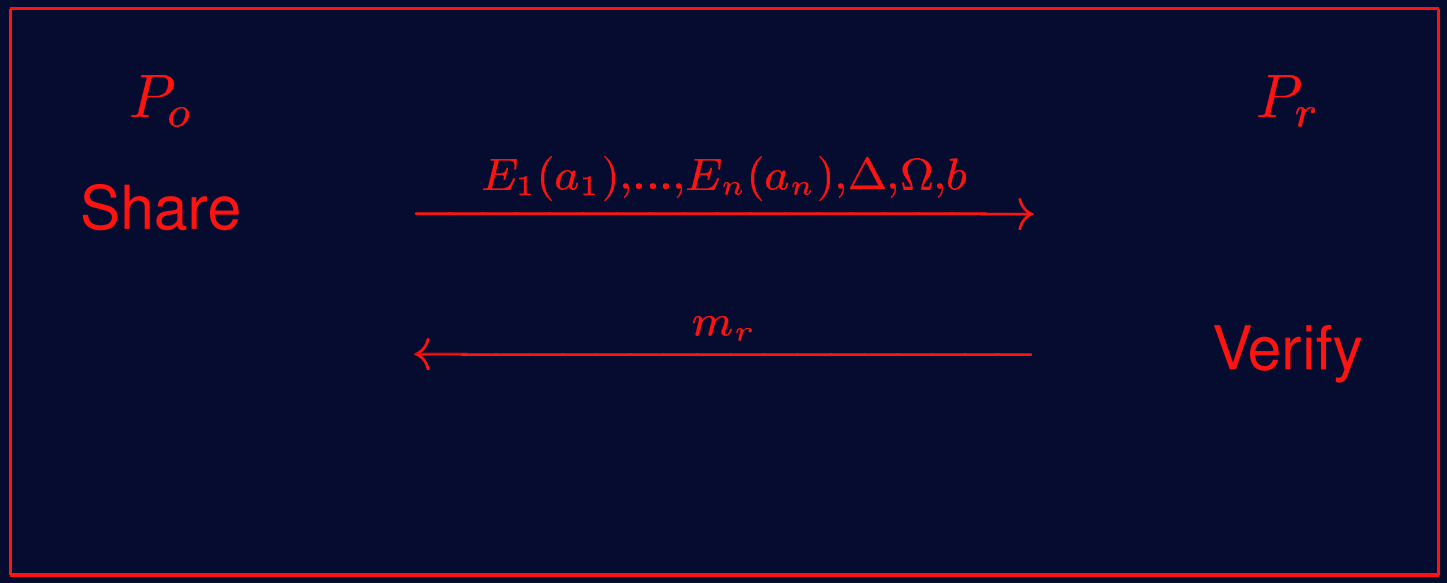
Share	Check	CheckEnc
Verify		Enc
Recover		Dec

Exchange protocol *Step by Step*



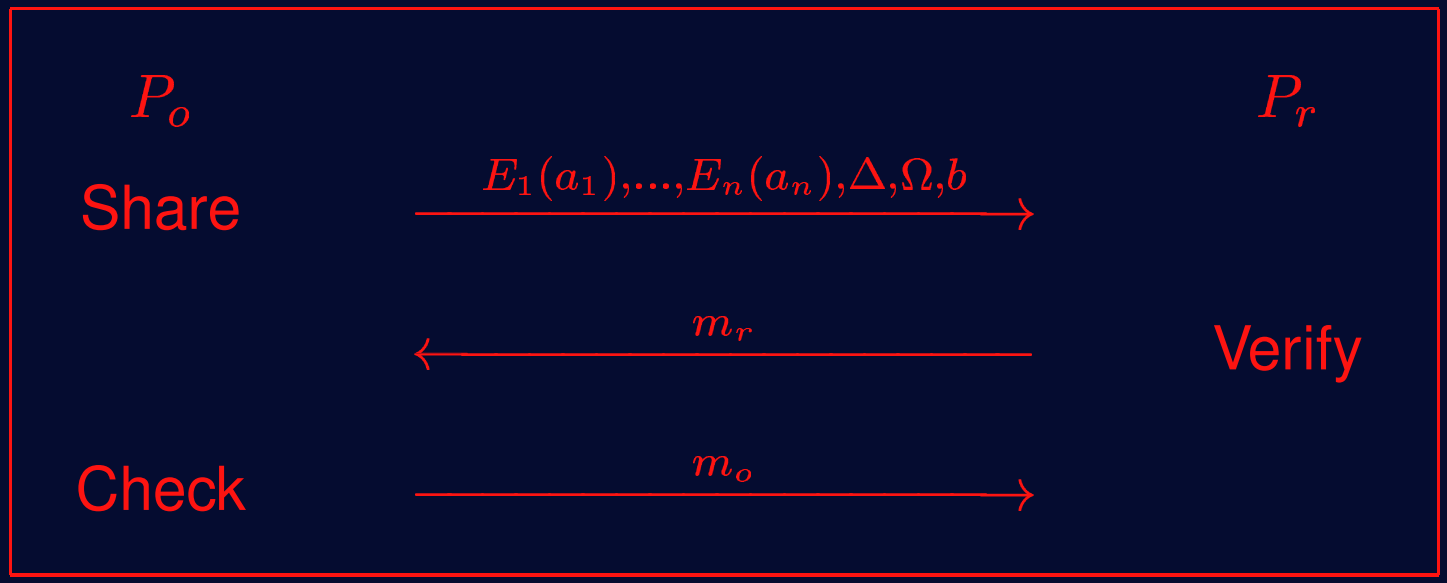
P_o picks a random element a and computes b such that $m_o = a + b$. He computes Share$(a, P_1, ..., P_n)$ and sends $E_1(a_1), ..., E_n(a_n), \Delta, \Omega, b$ to P_r.

Exchange protocol *Step by Step*



P_r checks that Verify$(E_1(a_1), ..., E_n(a_n), P_1, ..., P_n, \Delta,$ descr$(a))$ is true where descr(a) is deduced from descr(m_o) (e.g. $g^a = g^{m_o} g^{-b}$) and b. If the test succeeds then he sends m_r to P_o, else he has just to wait until the expiration date T to give up the exchange.

Exchange protocol *Step by Step*



P_o checks that m_r is correct running Check$(m_r,$ descr$(m_r))$. If it is the case then P_o sends m_o to P_r. Otherwise, he has just to wait until the expiration date T in order to give up the exchange.

Exchange protocol *Step by Step*

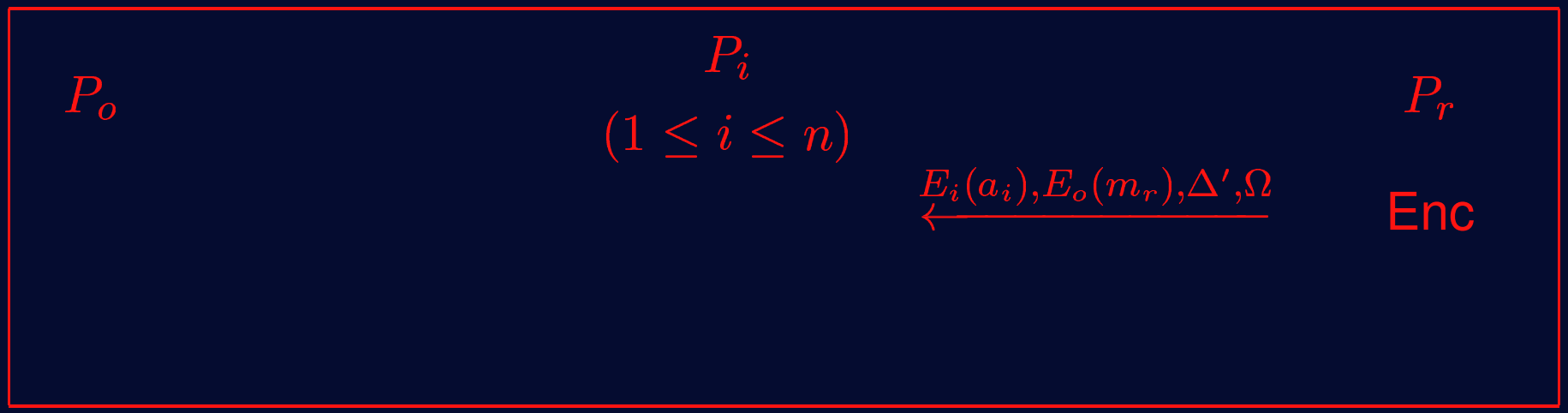
P_o		P_r
Share	$\xrightarrow{E_1(a_1),\ldots,E_n(a_n),\Delta,\Omega,b}$	
	$\xleftarrow{m_r}$	Verify
Check	$\xrightarrow{m_o}$	Check

If P_r does not receive m_o or if Check$(m_o,$ descr$(m_o))$ is false then he runs the recovery protocol.

Recovery protocol

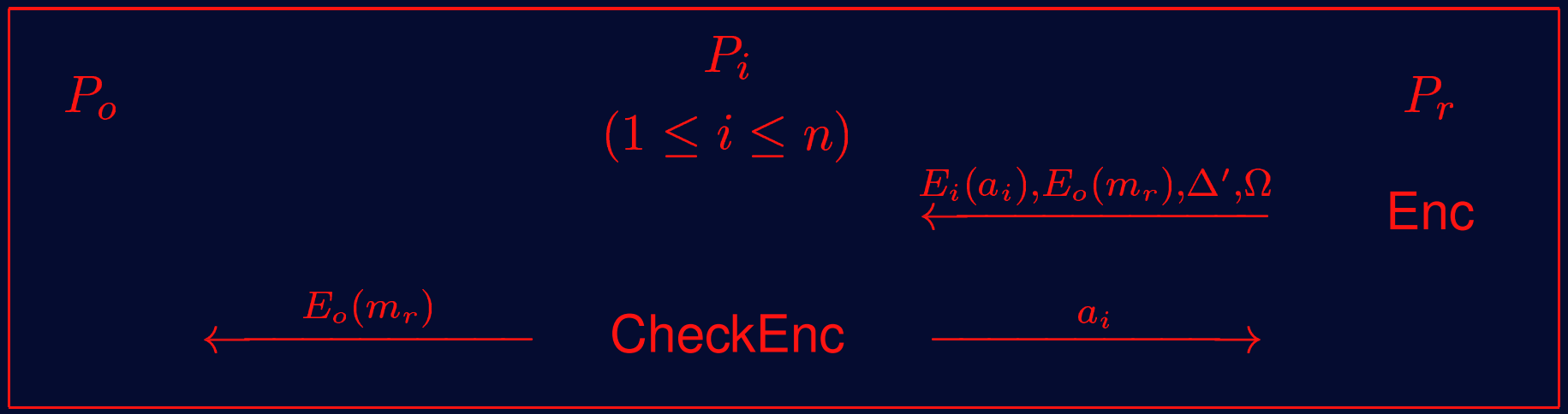
The recovery protocol is started before $T - T_{\text{max}}$ by the recipient, P_r, when he is injured, that is if the third message of the exchange, m_o, is wrong or missing.

Recovery protocol *Step by Step*



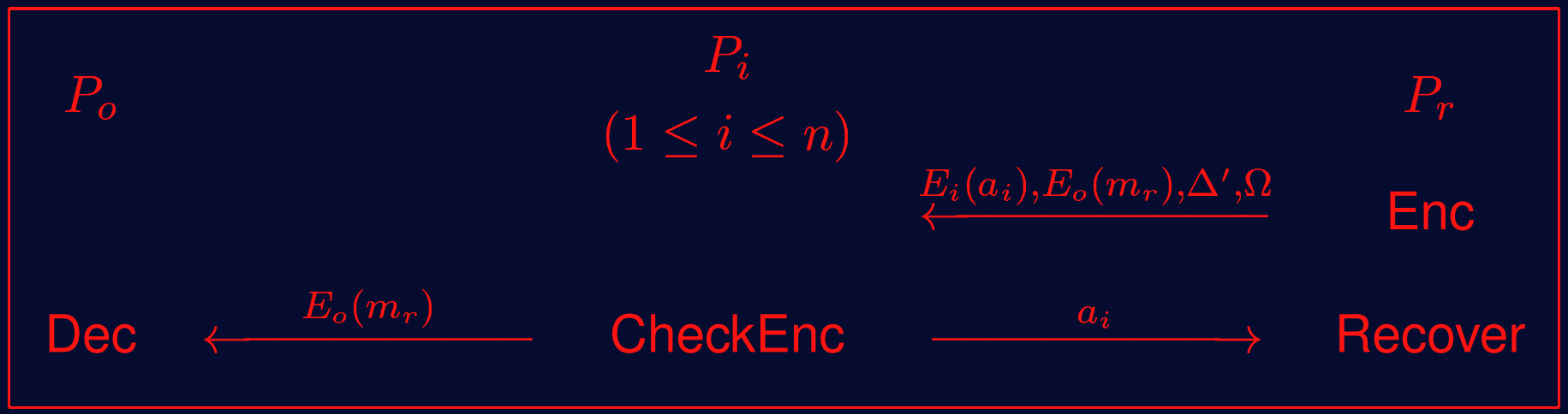
P_r encrypts m_r for P_o and sends $E_i(a_i)$, $E_o(m_r)$, Δ', and Ω to P_i.

Recovery protocol *Step by Step*



P_i computes CheckEnc$(E_o(m_r), \text{descr}(m_r), \Delta')$ where descr(m_r) is extracted from Ω; if the output is true and if the expiration date, contained in Ω, has not expired, P_i sends a_i to P_r and $E_o(m_r)$ to P_o.

Recovery protocol *Step by Step*



After having received k shares, P_r runs Recover. From a he computes $m_o = a + b$.

Analysis of the Protocol

Fairness

P_o is honest and P_r is dishonest.

- M_2 is wrong (or missing): P_o detects that M_2 is wrong, therefore he does not transmit M_3 and waits for T. If P_r does not run the recovery protocol then nobody can obtain anything valuable on the expected items and the exchange is trivially fair. If P_r starts the recovery protocol after T then he cannot obtain m_o since $|\mathcal{B}_2| < k$. If P_r starts the recovery protocol before T then P_o receives m_r from an external participant either in $\mathcal{B}_1$ or $\mathcal{B}_3$; therefore the protocol is fair iff P_r can obtain m_o that is if and only if $|\mathcal{B}_1| + |\mathcal{B}_2| \geq k$.

Fairness (Cont)

P_o is dishonest and P_r is honest.

- M_1 is wrong (or missing): P_r detects that M_1 is wrong and decides not to disclose m_r.The exchange ends therefore on a trivially fair termination after T.

- M_1 is correct but M_3 is not (or missing): P_r detects such a wrong M_3 and therefore starts the recovery protocol. The fairness of the exchange relies thus on the ability of the passive participants to supply a to P_r, that is if and only if $|\mathcal{B}_1| + |\mathcal{B}_2| \geq k$. The fairness is so ensured since P_o has already received m_r in M_2.

Privacy

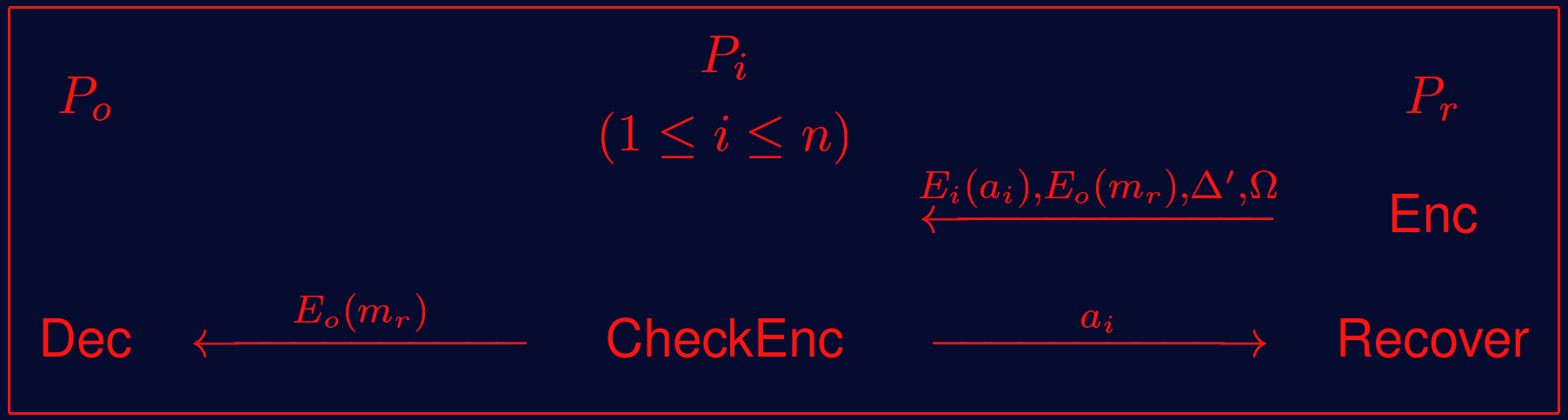
- If the recovery protocol is not performed, then only information between P_o and P_r are exchanged and passive participants receive nothing.

- If the recovery protocol is used, then some participants receive shares of a_i. However, although k participants colluding can recover a, they cannot recover m_o since they do not know b. Obviously, they cannot discover m_r either.

Complexity

When both P_o and P_r are honest, the complexity in terms of exchanged messages is very small since only three messages are exchanged.

When somebody misbehaves, the n passive participants are contacted by P_r, each receives one message and sends at most two messages, so the complexity (worst case) is only $O(3n)$.



Conclusion

Conclusion

- First optimistic fair exchange protocol which does not rely on a centralized trusted third party.

- We proved that our protocol ensures fairness even in quite dishonest environment and implies only low communication overhead.

- Our protocol works assuming that a majority of participants are honest or that only one is honest but we can estimate the number of participants in $\mathcal{B}_2$ who may harm P_o by colluding with P_r.