Does Parallel Repetition Lower the Error in Computationally Sound Protocols?

Mihir Bellare
UC San Diego

Russell Impagliazzo
UC San Diego

Moni Naor
Weizmann Institute

מכון
ויצמן
למדע

WEIZMANN
INSTITUTE
OF SCIENCE

Alice and Bob play a game.

Alice prob of winning the game is at most ε.

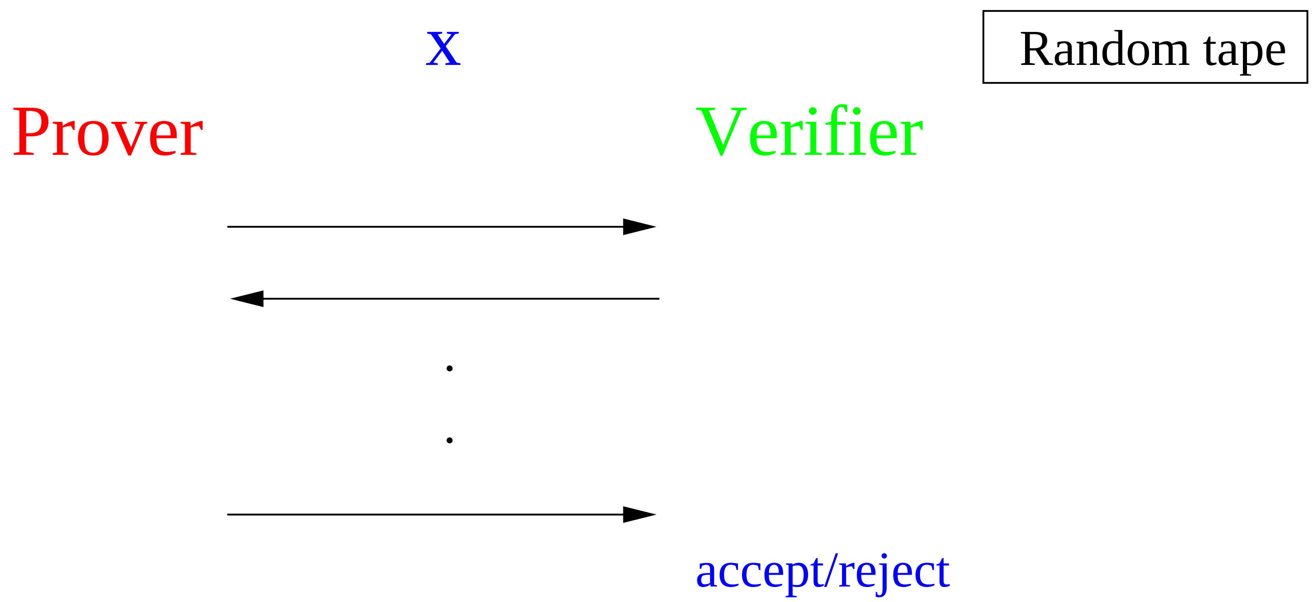
What are Alice's chances of winning k games?

- When the games are played sequentially
- When the games are played concurrently

Applications:

- Cryptographic Protocols
- Automated Turing test

Typical case



Protocol defined by a verifier V who at end either accepts or rejects the common input x.

Typical conditions:

– **Completeness:** Good prover can make V accept a true claim

– **Soundness:** Even a bad prover can seldom make V accept a false claim.

Soundness

Statistical soundness: Interactive proofs [GMR] and Arthur Merlin games [B].

- Soundness holds even for a computationally unbounded prover

Computational soundness: Arguments [BC] and CS proofs [M]

- Soundness holds ONLY for a computationally bounded prover.

For simplicity take the argument setting where computationally bounded means polynomial time.

A large number of cryptographic protocols are arguments.

The Error

The error probability ϵ of the protocol defined by V is the chance that a bad prover can make V accept a false claim.

The error is ϵ if:

In statistical soundness setting: For every $\widehat{P}$ and every false claim x the chance that $\widehat{P}$ can make V accept is at most $\epsilon(|x|)$.

In computational soundness setting: For every polynomial time $\widehat{P}$ and long enough false claim x, the chance that $\widehat{P}$ can make V accept is at most $\epsilon(|x|)$.

For emphasis sometimes call latter computational error.

Error Reduction

Many/Most "natural" protocols are atomic: they have constant error like 1/2. We need to lower it. This process is called error reduction or amplification.

Two main kinds of generic error-reduction techniques: serial repetition and parallel repetition.

In either case, the new verifier runs k independent copies of V, and accepts iff V accepts in all sub-protocols.

We say the error decreases at an *exponential rate* if poly($k(n)$) repetitions yields $2^{-k(n)}$ error for "appropriate" $k(n)$.

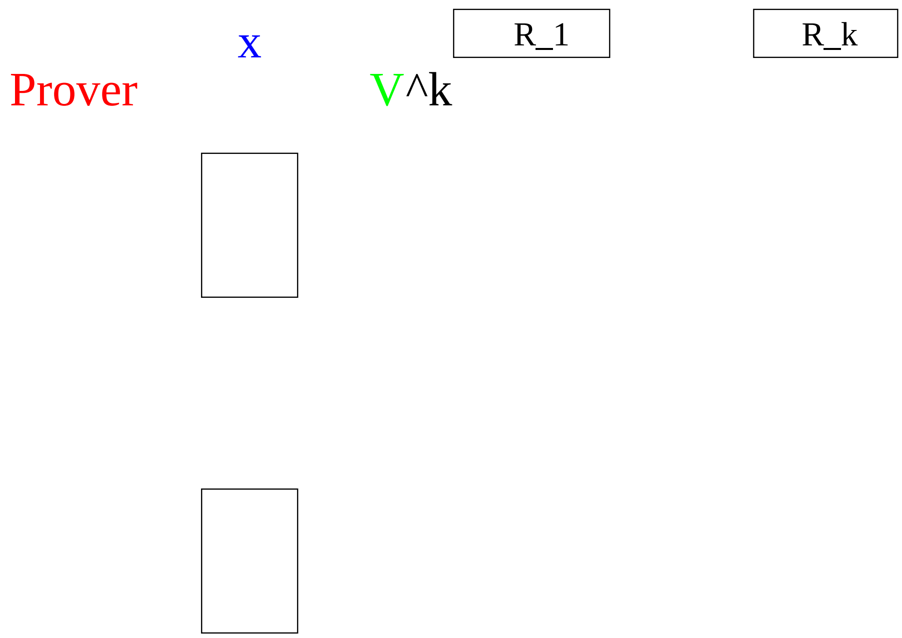
The Danger

The bad prover's moves in one sub-protocol can depend on those in another!

It may return answers as a function of all known information, co-relating across different sub-protocols.

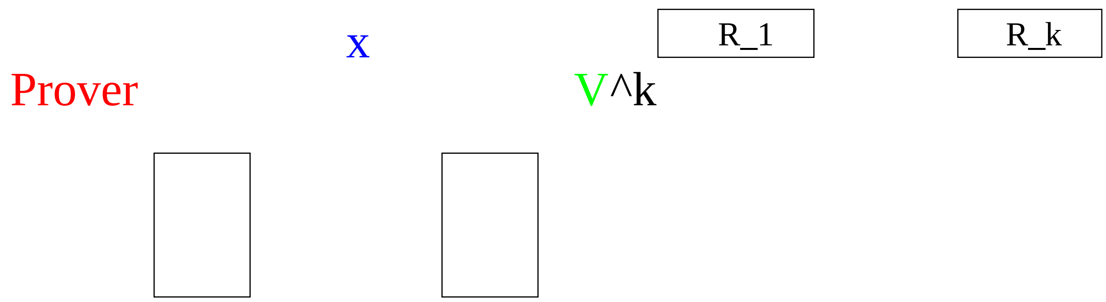
This is what makes repetition non-trivial.

Serial Repetition



- Known to reduce error at an exponential rate in all reasonable models
- Even preserves zero-knowledge

Parallel Repetition



For statistically sound games error reduces exponentially!

Proof: consider the game-tree...

Zero-knowledge

- Is not preserved [GK]!
- A lot of work to get parallel zk (modify alg.)
- Concurrent executions? (Recent work)

Important question for multi-prover proofs. Here:

- Repetition might not lower the error at the "obvious" rate [Fort]!
- Two repetitions might not lower error at all [Fe]
- Deep result: For every protocol there is α s.t. $k(n)$ reps. lowers it to $\alpha^{-k(n)}$ [Ra]
- This is best possible [FV].

Computational Error Reduction

Does parallel repetition reduce the error in a computationally sound setting?

More precisely:

- Given V with constant (computational) error
- Given $k(\cdot)$
- What is error of V^k?

In particular: Does parallel repetition lower the error in arguments?

Surprisingly, the answer to these questions was not known/considered.

What's the problem?

Surely parallel repetition lowers error at an exponential rate even in computationally sound protocols and this should not be too hard to show.

(1) Doesn't this follow from the result about interactive proofs (stat. soundness)?

- No, because the starting protocol is only computationally sound, a weaker property.

(2) Should be an exercise

- Try it.

Summary of our results

Negative: Parallel repetition in general does not lower the error in a computationally sound setting. We show protocols for which:

- Two repetitions fail to lower error
- For any k there is a protocol where k repetitions fail to lower error
- No *black box transformation* can show $k(\cdot)$ repetitions lowers the error, for any $k(\cdot) = \text{poly}(\cdot)$.

These protocols have four rounds.

Positive: Error goes down at an exponential rate in three round protocols.

Significance

Parallel repetition can be / is used to lower error in witness hiding protocols [FeSh1]. Variants of parallel repetition (that preserve ZK) are used extensively.

Many atomic protocols are three rounds and our positive result applies.

But need to be careful about computational soundness claims for protocols of more than three rounds: it will typically have to be argued on a case by case basis (eg. [FeSh2,BJY]).

In some protocols in literature, soundness seems to be "assumed".

Automated Turing Test

Motivation: Want to prevent softbot from using resources.

Idea: To enter a site provide a challenge in which humans excel but even the best programs are not too successful (NLP, gender recognition, line drawings, ...?)

If basic problem has constant prob. of success for program, does repeating the challenge make it harder? *Not completely obvious...*

Research direction: come up with a *public* automated Turing test.

Another application: combatting junk mail.

Non-malleable encryption [DDN]

- Encryption: $B = \mathcal{E}_{pk}(b, r)$, bit b and coins r
- Decryption: $b = \mathcal{D}_{sk}(B)$

Computationally infeasible task:

$$B \;\rightarrow\; B' \text{ such that } \begin{cases} B \neq B' \text{ but} \\ b' \text{ is “related” to } b\,, \end{cases}$$

where

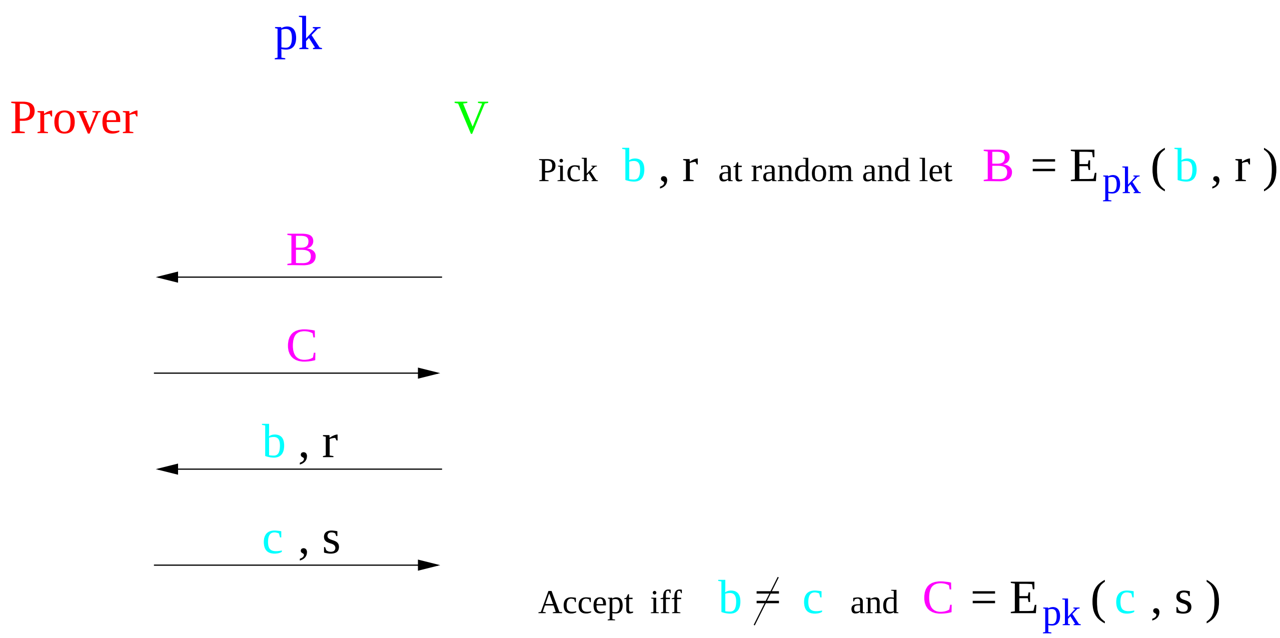
- $b' = \mathcal{D}_{sk}(B')$ and $b = \mathcal{D}_{sk}(B)$

In particular, given $B = \mathcal{E}_{pk}(b, r)$ we can't construct an encryption of $\bar{b}$ or a different encryption of b.

NM-Security is stricter than *Semantic Security*.

Theorem [DDN]: Non-malleable encryption schemes exist if trapdoor permutations exist.

Protocol DD



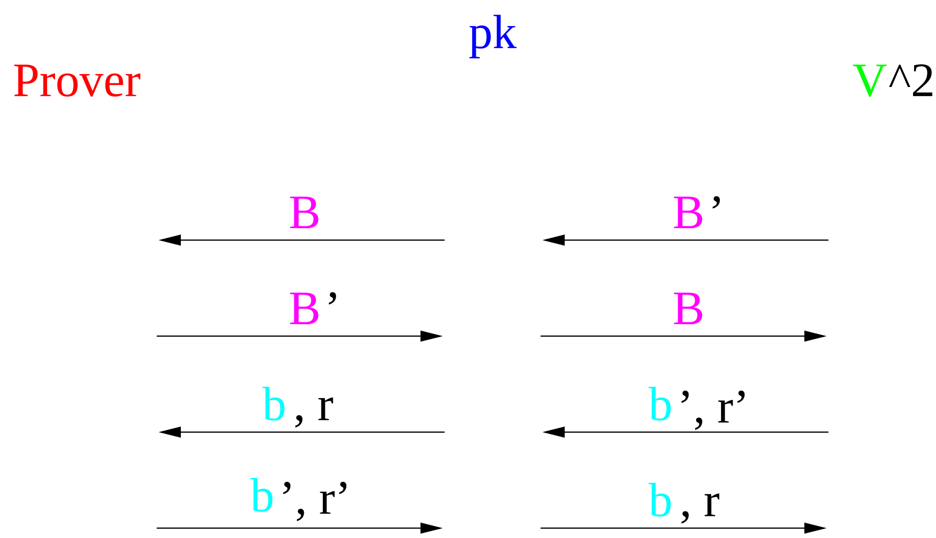
Claim 1: DD has computational error $1/2$, assuming the encryption scheme is non-malleable.

DD is a "natural" protocol for proving *computational ability*. However...

Doesn't parallelize:

Claim 2: There is a polynomial time strategy for a prover to convince the verifier with probability $1/2$ in DD^2.

How to cheat in DD^2



A strategy for a cheating prover:e cheater just copies stuff across sub-protocols.

V^2 accepts if (returned ciphertexts are correct) *and* ($b \neq b'$ and $b' \neq b$), which happens with probability $1/2$!

What's next?

Can generalize for any k - there is a DD_k where the error in DD_k^k does not reduce.

This still leaves the possibility that for any protocol enough (parallel) repetitions lower the error (at exponential rate).

Will show — unlikely as well.

First discuss the positive result.

Will lead up to this via a closer look.

Computational error can't get too low

In typical computationally sound protocols, prover can "cheat" if it manages to solve some hard problem like factoring or discrete log.

If this happens with prob. $\mu(n)$, then the error cannot be expected to go below this, however often we repeat.

In particular, since $\mu(n)$ is usually only negligible, we don't expect the computational error for $k(\cdot) = \text{poly}(\cdot)$ repetitions (whether serial or parallel!) to be $2^{-k(\cdot)}$.

This already indicates a difference in comparison with interactive proofs.

Reductionist approach

Pretty much the only imaginable route to showing a claim like

"If V has error ϵ then V^k has error δ"

is by a reduction:

$$\boxed{\text{Given } P^* \text{ who makes } V^k \text{ accept with prob.} > \delta\text{, we construct } \widehat{P} \text{ who makes } V \text{ accept with prob.} > \epsilon.}$$

Remember δ is small and ϵ is big!

Furthermore, transformation is typically black box:

$$\widehat{P} = M^{P^*}$$

for some polynomial time algorithm M.

Direct Product

The “right” analogue for our question is direct product questions [Y].

Eg: Weak one-way function → Strong one-way function.

Given f define

$$f^{\mathrm{poly}(k)}(x_1, \ldots, x_{\mathrm{poly}(k)}) = f(x_1), \ldots, f(x_{\mathrm{poly}(k)}) \; .$$

Can show: if prob. of inverting f in poly-time is $1/2$, then prob. of inverting $f^{\mathrm{poly}(k)}$ in poly-time is

$$\max(1/\mathrm{poly}(n), 2^{-k}).$$

Shown via a reduction: given an inverter for $f^{\mathrm{poly}(k)}$ construct an inverter for f.

Can’t we apply this to our setting?

Difficulties

Difficulty in applying direct product techniques to our setting is interaction and private coins of verifier.

Given: P^* who makes V^k accept with prob. $> \delta$
Want: $\widehat{P} = M^{P^*}$ who makes V accept with prob. $> \epsilon$.

$\widehat{P}$ receives a message C from V and must find a reply A. It can use P^* to get an answer but must estimate the quality of this answer on-line before sending it.

This can be addressed for three rounds via the trust halving strategy of [IW], but breaks down for four or more rounds.

Card guessing game [IW]

Casino Xor:

- k card $x_1, \ldots x_k$ faced down.
- Arbitrary distribution such that

$$\Pr[\text{for all } 1 \leq j \leq k \; X_j = 1] = p.$$

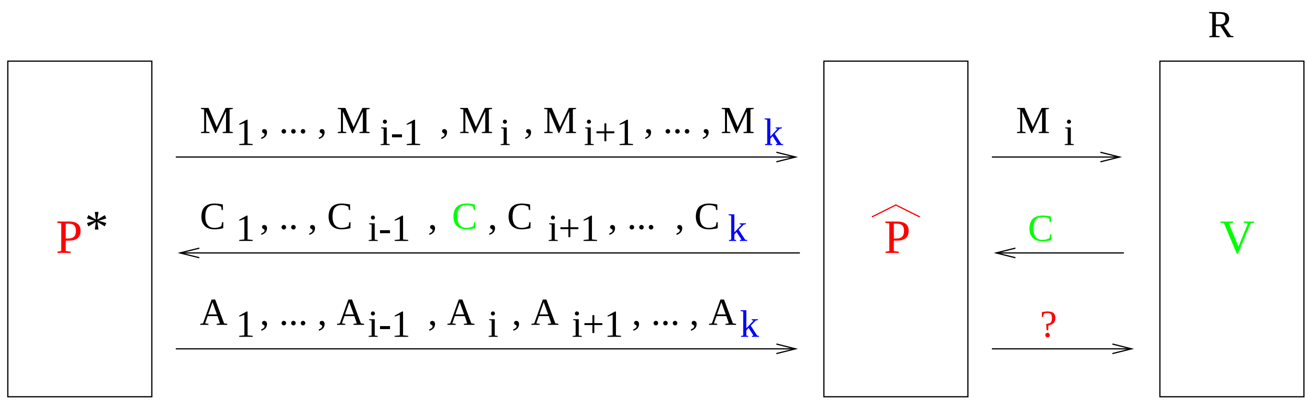
- $1 \leq I \leq k$ random. For all $j \neq I$ card x_j turned up.
- Player can wage that $x_I = 1$ (or not).

(Almost) Optimal Strategy (THS):

- Let t be number of 0 cards.
- Accept wager with Prob. 2^{-t}.

Theorem [IW]: For any dist. on $x_1, \ldots x_n$ which is 1^k with prob. at p, THS has expected payoff at least $p - 2^{-k/3}$.

Using the Trust Halving Strategy (THS)



Prover $\widehat{P}$:

- Picks coins $R_1, \ldots, R_{i-1}, R_{i+1}, \ldots, R_k$ for V
- Runs P^*, then feeds M_i to V
- Gets "challenge" C from V
- Determines $C_j = V(M_j, R_j)$ for $j \neq i$ and sends vector
- Would like to return A_i for ?

Uses A_i with probability 2^{-t} where t is the number of rejecting sub-protocols amongst

$$1, \ldots, i-1, i+1, \ldots, k\ .$$

Else try again.

Positive result

Theorem: Let V define a three round verifier strategy with computational error probability $\epsilon = 1 - \mu(\cdot)$, and let $k = k(n)$ be $O(\log n)$. Then V^k has error probability

$$\delta(\cdot) = \frac{32}{\mu(\cdot)} \cdot e^{-\mu(\cdot)^2 \cdot k/128} .$$

Furthermore the reduction is black box.

Four round protocols

For protocols of four or more rounds, there is still the possibility of a Raz-like result:

For any protocol there is α there is α s. t. k repetitions lower error to α^{-k}.

However, we now indicate that even this is unlikely.

Provide a protocol for which after any poly($\cdot$) repetitions the error can't be shown to be lower via Black box amplification.

Direct Product

Theorem: Assume non-malleable encryption schemes exist. Then there is a protocol LC having error a constant ϵ such that for any polynomial $k(\cdot)$, there is no black box reduction that can show that the error of LC^k drops below ϵ.

Protocol LC

Common input: pk_1, pk_2

- V picks a random trit $b \in \{0, 1, 2\}$ and coins r. It sets $B = \mathcal{E}_{pk_1}(b, r)$ and sends B to the prover
- The prover sends a ciphertext C to V
- V sends b, r to the prover
- The prover sends a trit $c \in \{0, 1, 2\}$ and a string s to V
- V accepts iff $\mathcal{E}_{pk_2}(c, s) = C$ and $b + c = 0 \bmod 3$.

Intuition

Why is there no black box amplifier?

Oracle:

Input $C_1, \ldots C_m$ where $C_i = \mathcal{E}_{pk_1}(x_i; t_i)$ for $i = 1, \ldots, m$:

- Check that $C_i \neq C_j$
- Set $x = x_1 + \ldots + x_m$ mod 3
- Return $C = \mathcal{E}_{pk_2}(x; s)$

Claim: With access to such an oracle the error of LC^k is 1/3.

Claim: Even with access to such an oracle the error of LC is at most 1/3 (plus negligible function).

Further Research

On-going work: We (Bellare, Halevi, Naor) can prove similar negative results for proofs of knowledge.

Concurrent Zero-knowledge: using *timing* assumptions [DNS].

Open: What happens to protocols which are Arthur-Merlin?

Open: Are there general and reasonable constraints on protocols that imply error reduction

More information: Full paper available via

http://www-cse.ucsd.edu/users/mihir

Extended abstract appeared in FOCS 97.