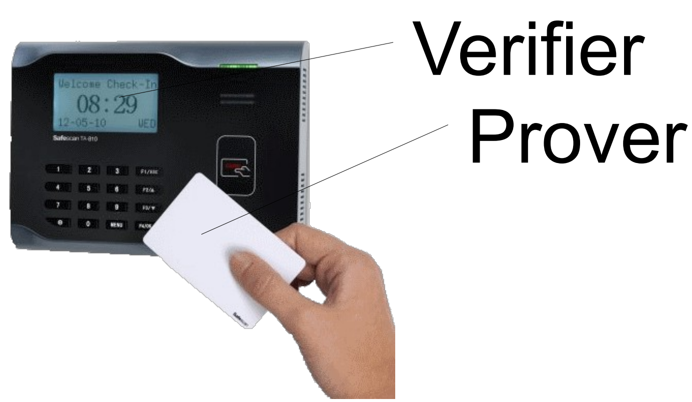
Distance bounding protocols

If you think cryptography
is the answer to your problem,
then you don't know what your problem is.

Peter G. Neumann,
quoted in the New York Times, February 20 2001.

The “mafia” fraud

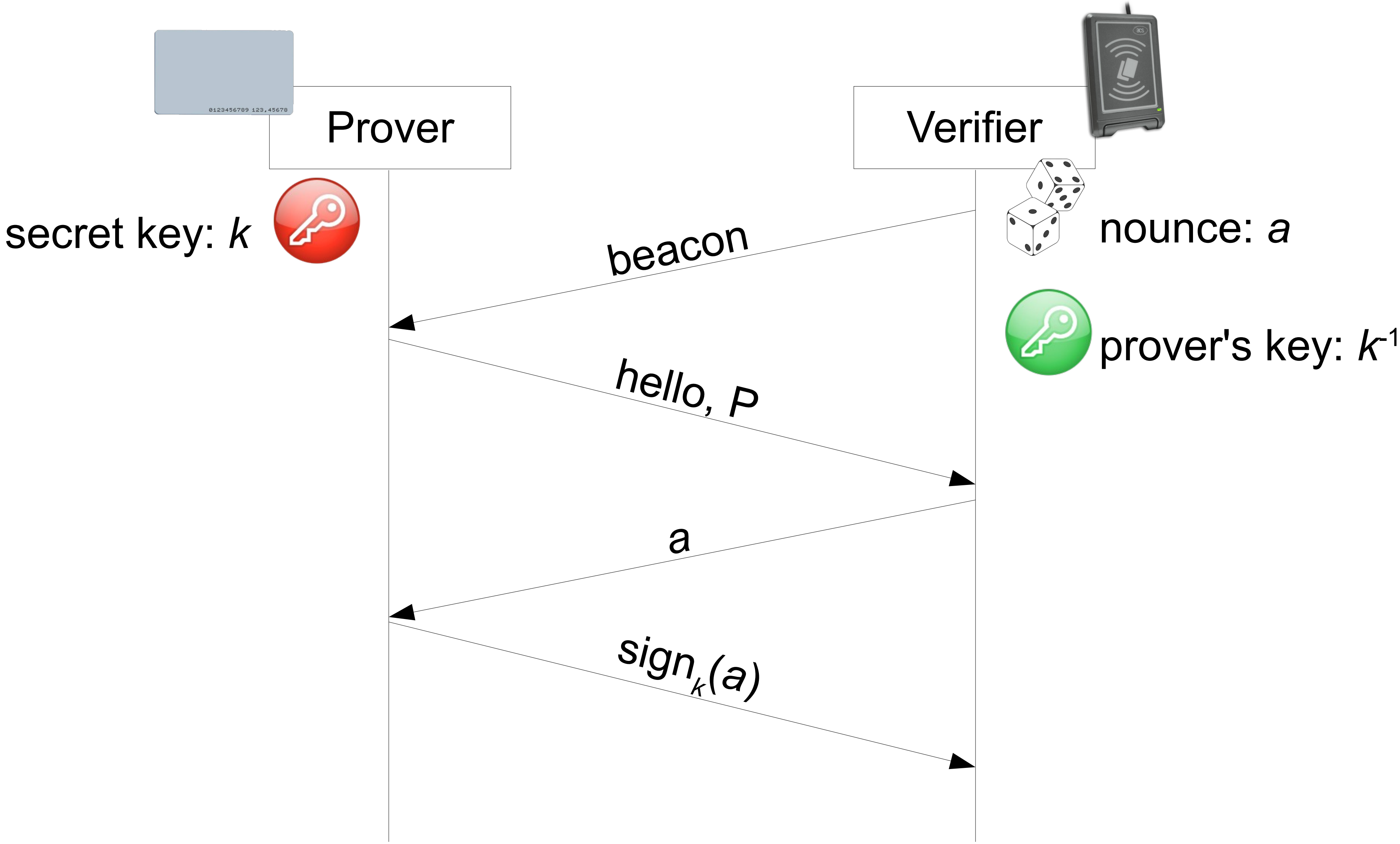
- The top-secret area contains big military secrets (crashed UFOs, mind-control technologies, etc.)
- The "men in black" employees access the top-secret area with a contactless smart card



The “mafia” fraud

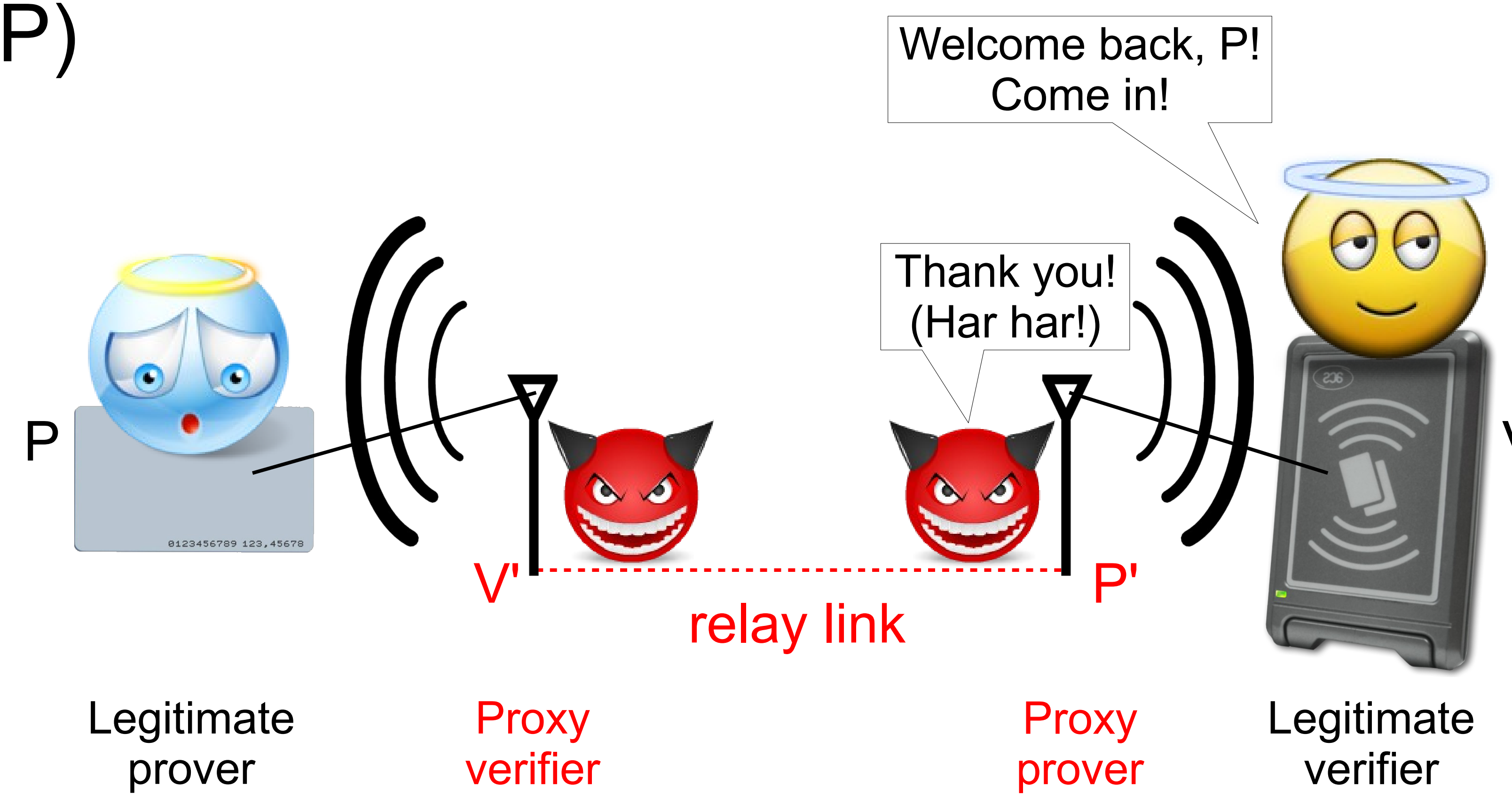
- Suppose that:
  - The smart cards cannot be stolen (man in black are very professional)
  - The smart cards cannot be cloned (asymmetric cryptography with tamper-proofness)
  - The authentication protocol between verifier and prover is correct (BAN logic proof)
  - The employed crypto primitives are unforgeable (the cryptoanalyzer are good in maths)
- There is still a way to completely break the system

The “mafia” fraud

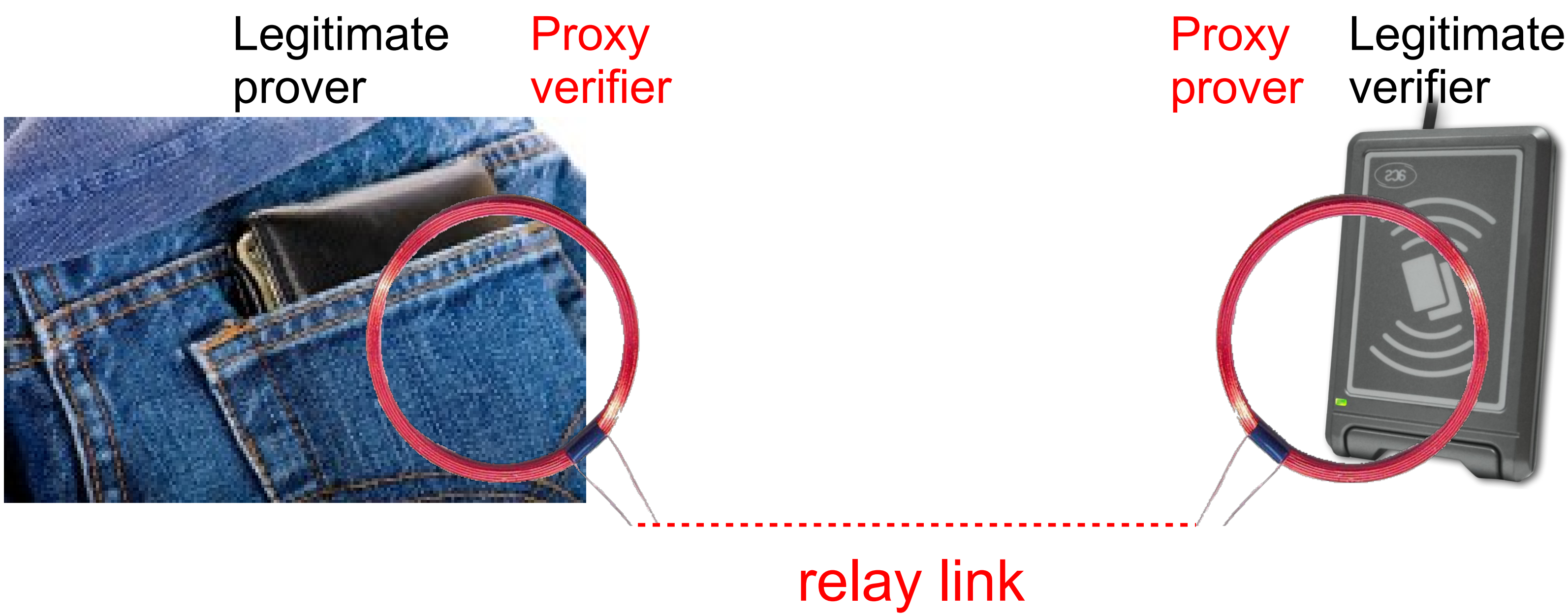


The "mafia" fraud

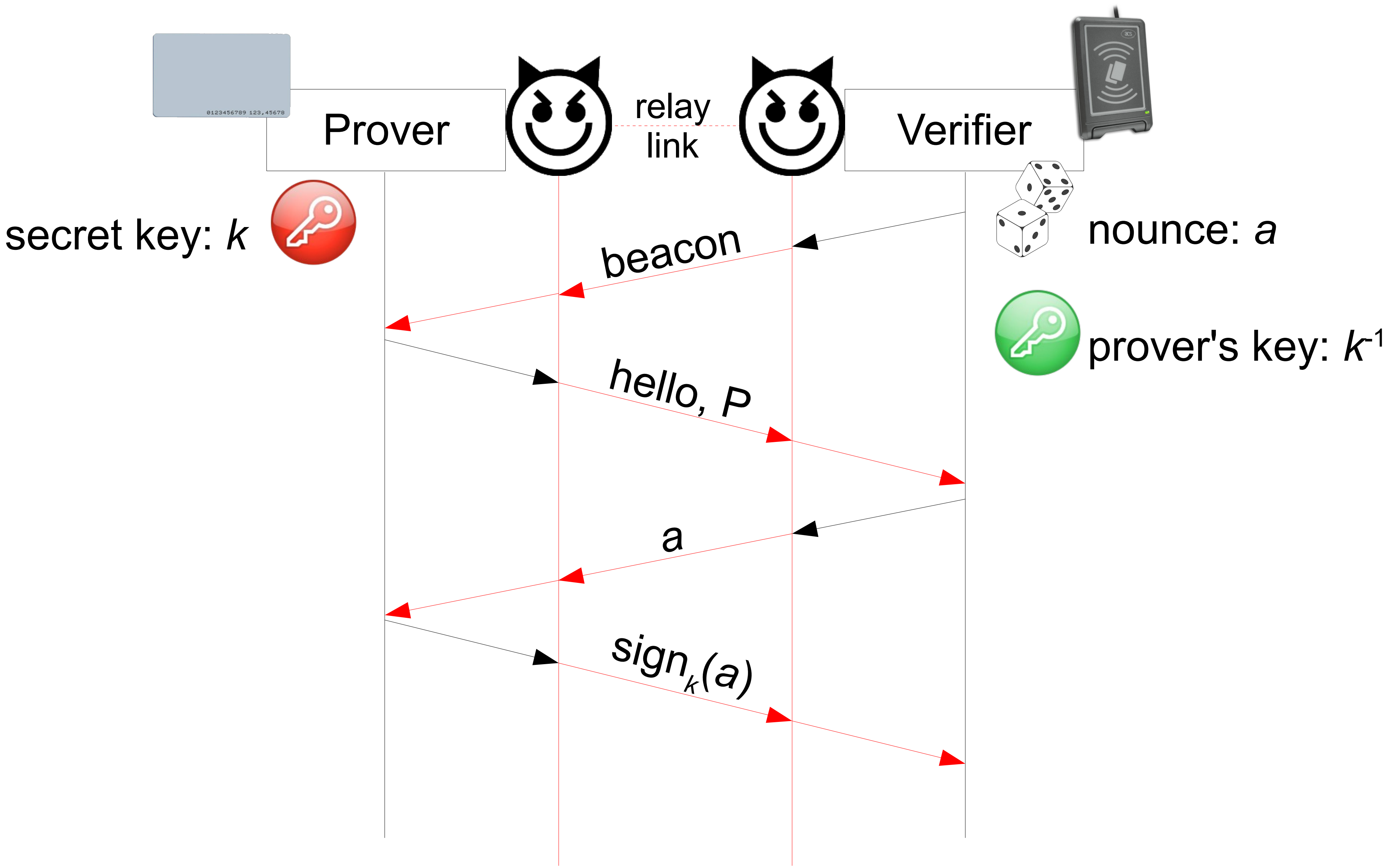
- Build a *relay link* (possibly an *Internet link*) which makes a legitimate verifier (V) communicate with a far away legitimate prover (P)



The “mafia” fraud



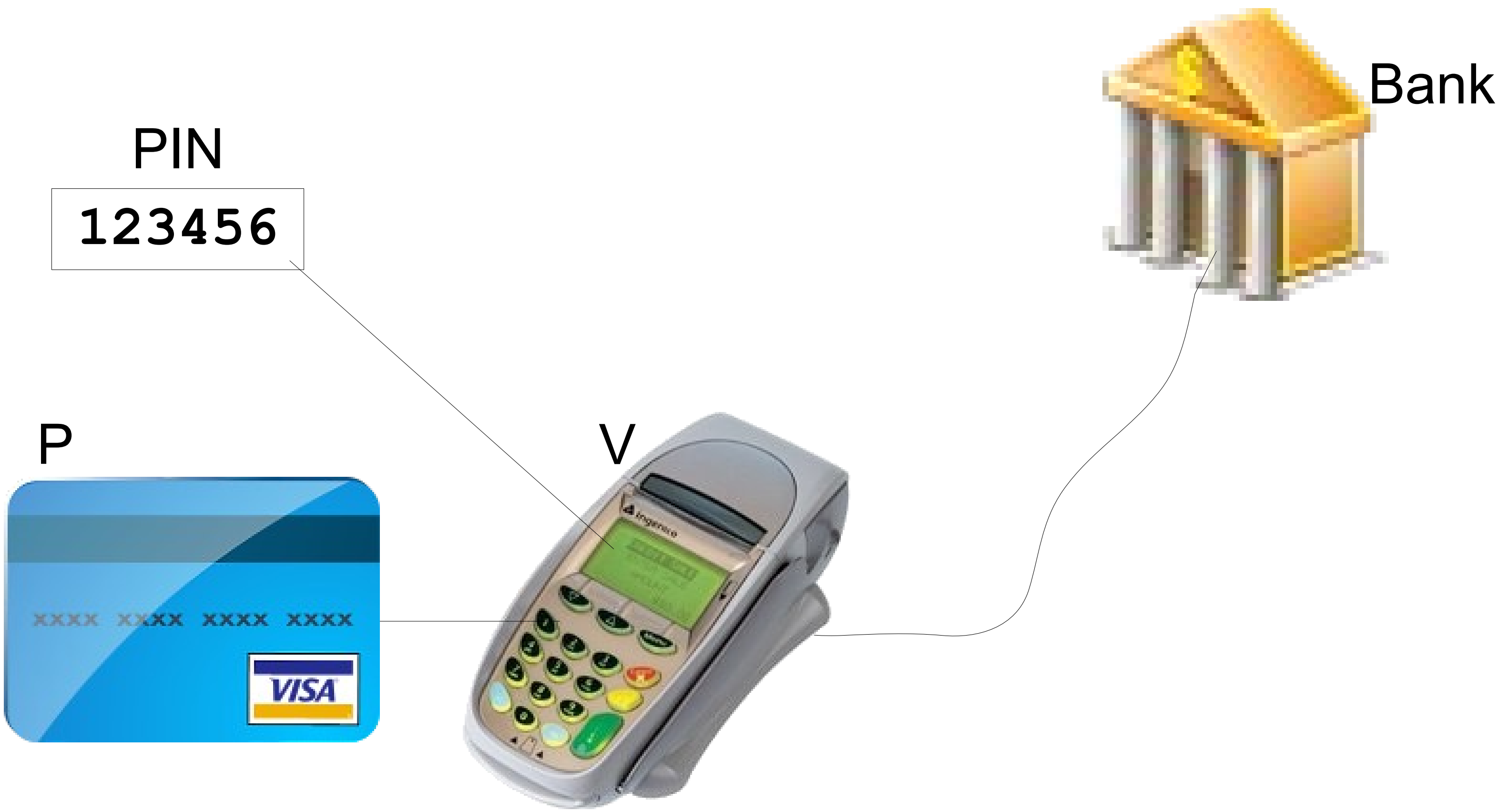
The “mafia” fraud



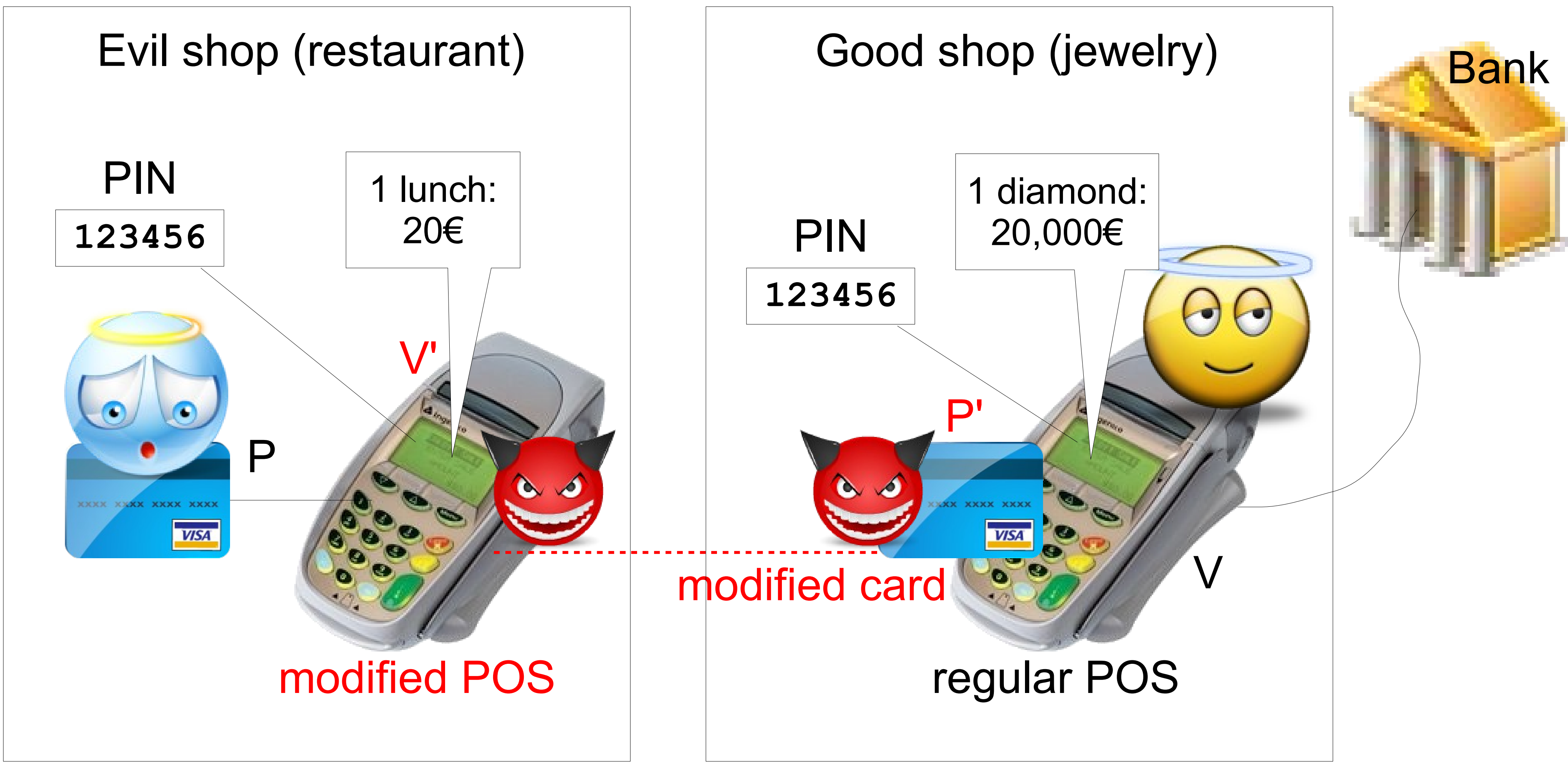
The “mafia” fraud

- *False assumption*: "If two devices can hear each other, then they are *close* to each other"
- Sometimes called "relay attack", "wormhole attack"
- Other examples: credit card payments, car stealing, wireless routing

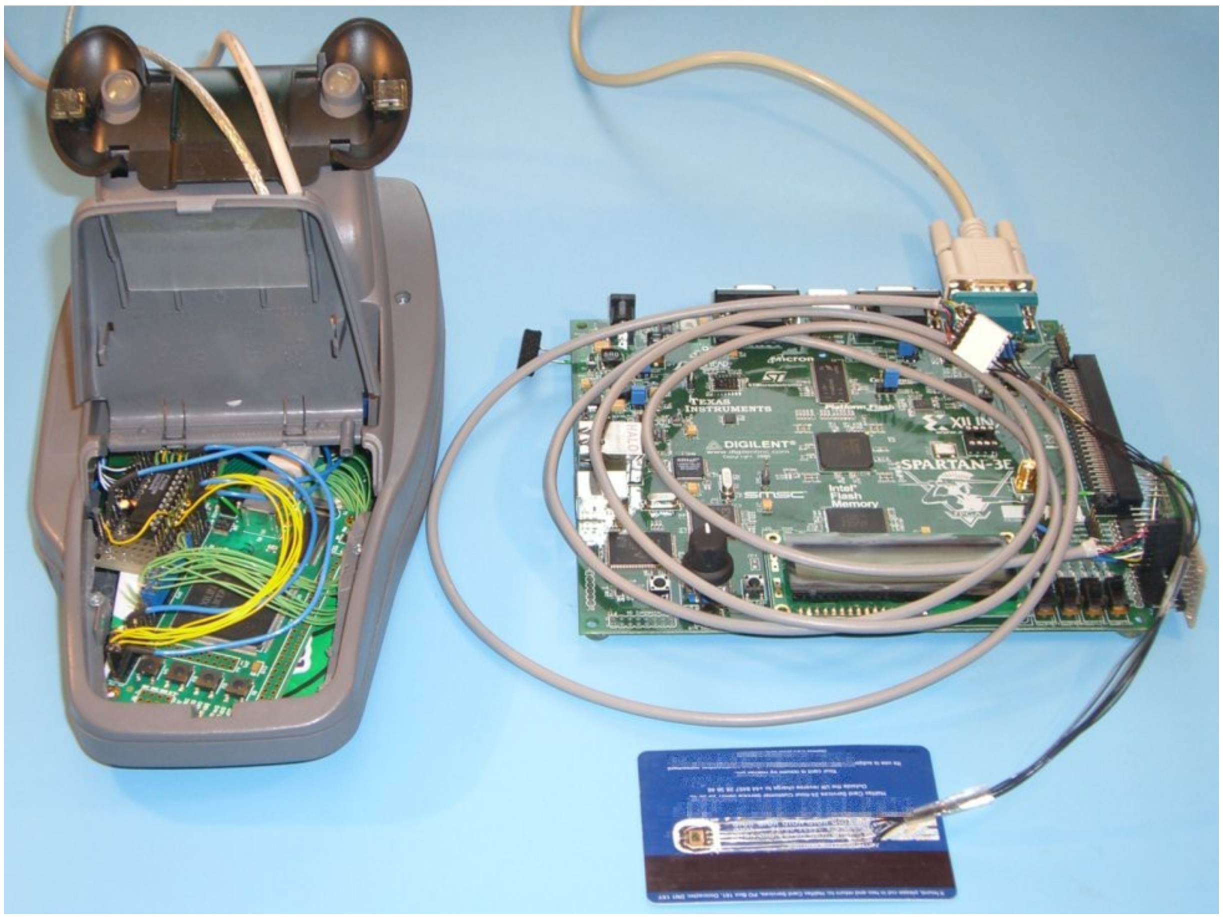
Mafia fraud against chip&pin payments



Mafia fraud
against chip&pin payments



Mafia fraud against chip&pin payments



Mafia fraud against PKES

- Passive Keyless Entry and Start

Mafia fraud against PKES

Mafia fraud against PKES

Car model	Relay cable					
	7 m		30 m		60 m	
	open	go	open	go	open	go
Model 1	✓	✓	✓	✓	✓	✓
Model 2	✓	✓	A	A	A	A
Model 3	✓	✓	✓	✓	✓	✓
Model 4	✓	✓	-	-	-	-
Model 5	✓	✓	✓	✓	✓	✓
Model 6	✓	✓	A	A	A	A
Model 7	✓	✓	A	A	-	-
Model 8	✓	A	✓	A	-	-
Model 9	✓	✓	✓	✓	✓	✓
Model 10	✓	✓	✓	✓	-	-

✓ Without amplification

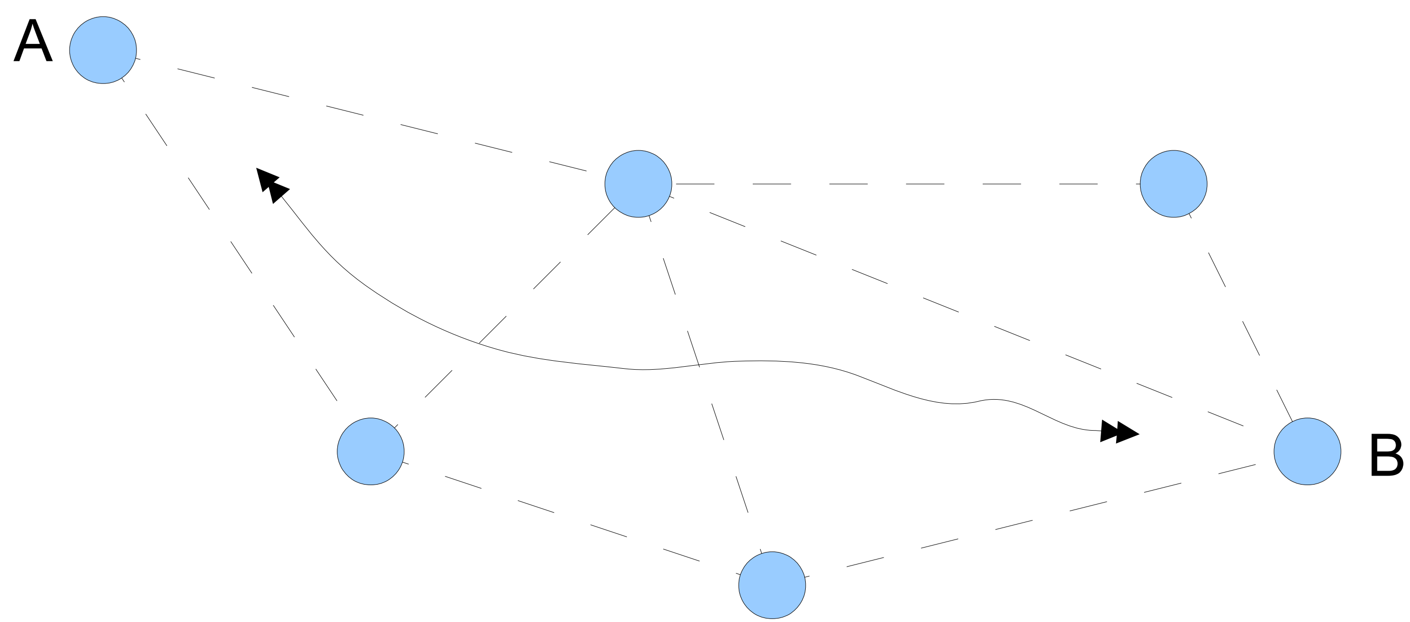
A With amplification

– Not tested

2010

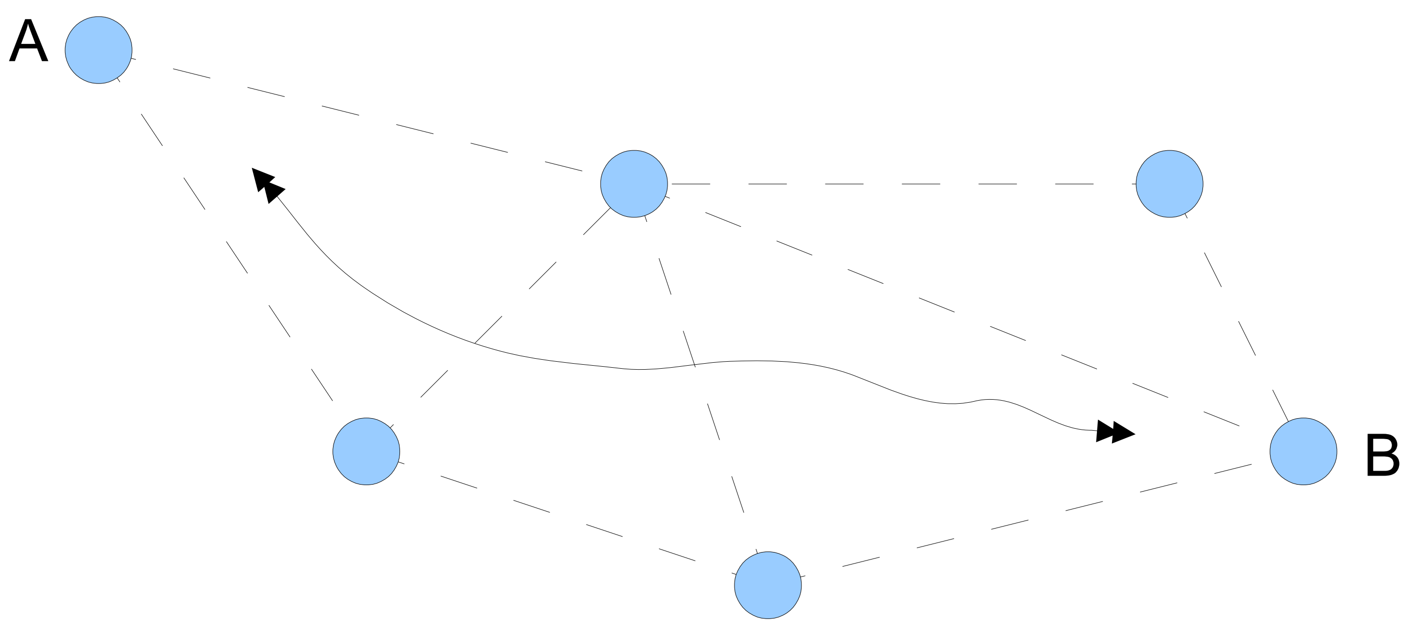
Wormhole attack

- False assumption: "if A hears an (authenticated) beacon message from B, then B and A are in the proximity"
- The adversary establishes a (wireless) link between two far away nodes (the wormhole)



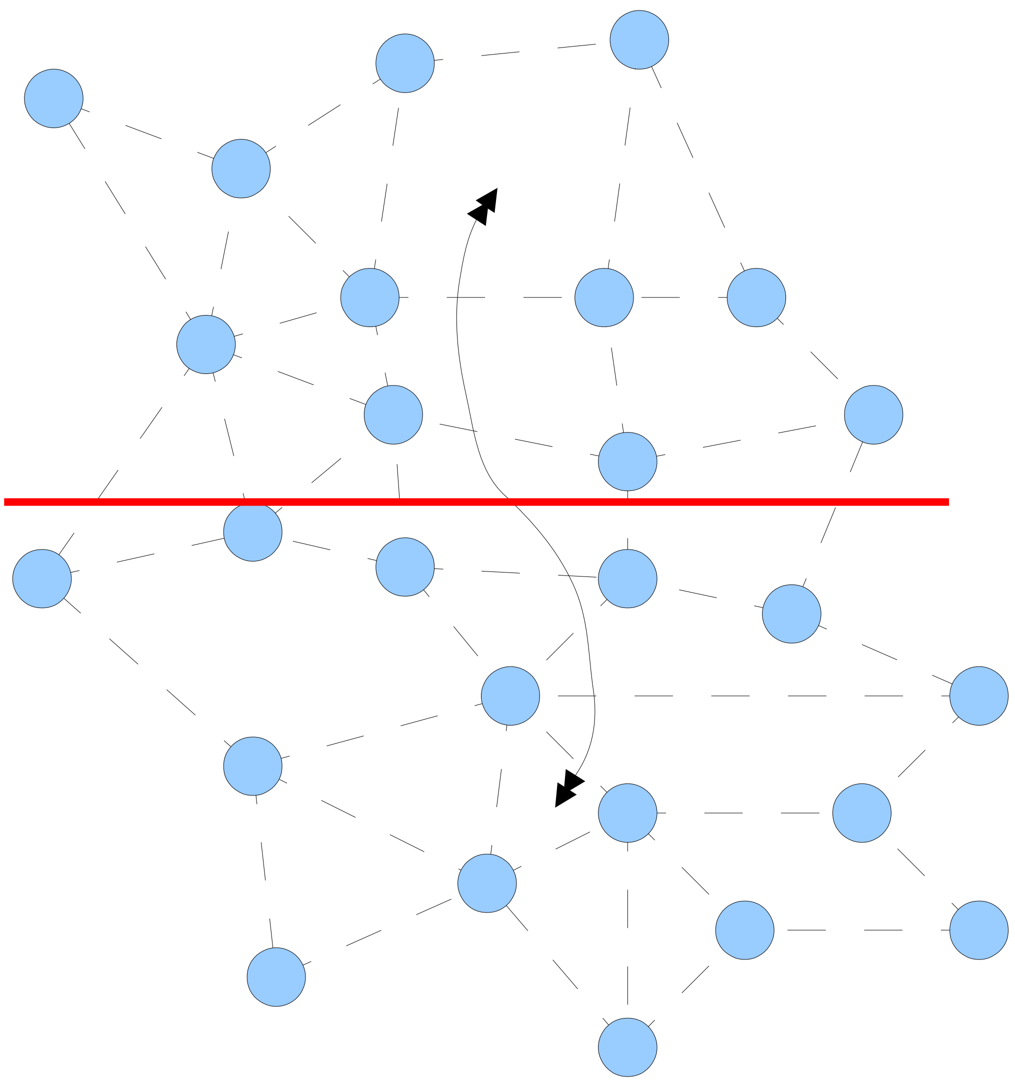
Wormhole attack

- A and B become *de facto* neighbours
- The wormhole is controlled by the adversary
- The adversary can suppress the traffic partially or totally



Wormhole attack

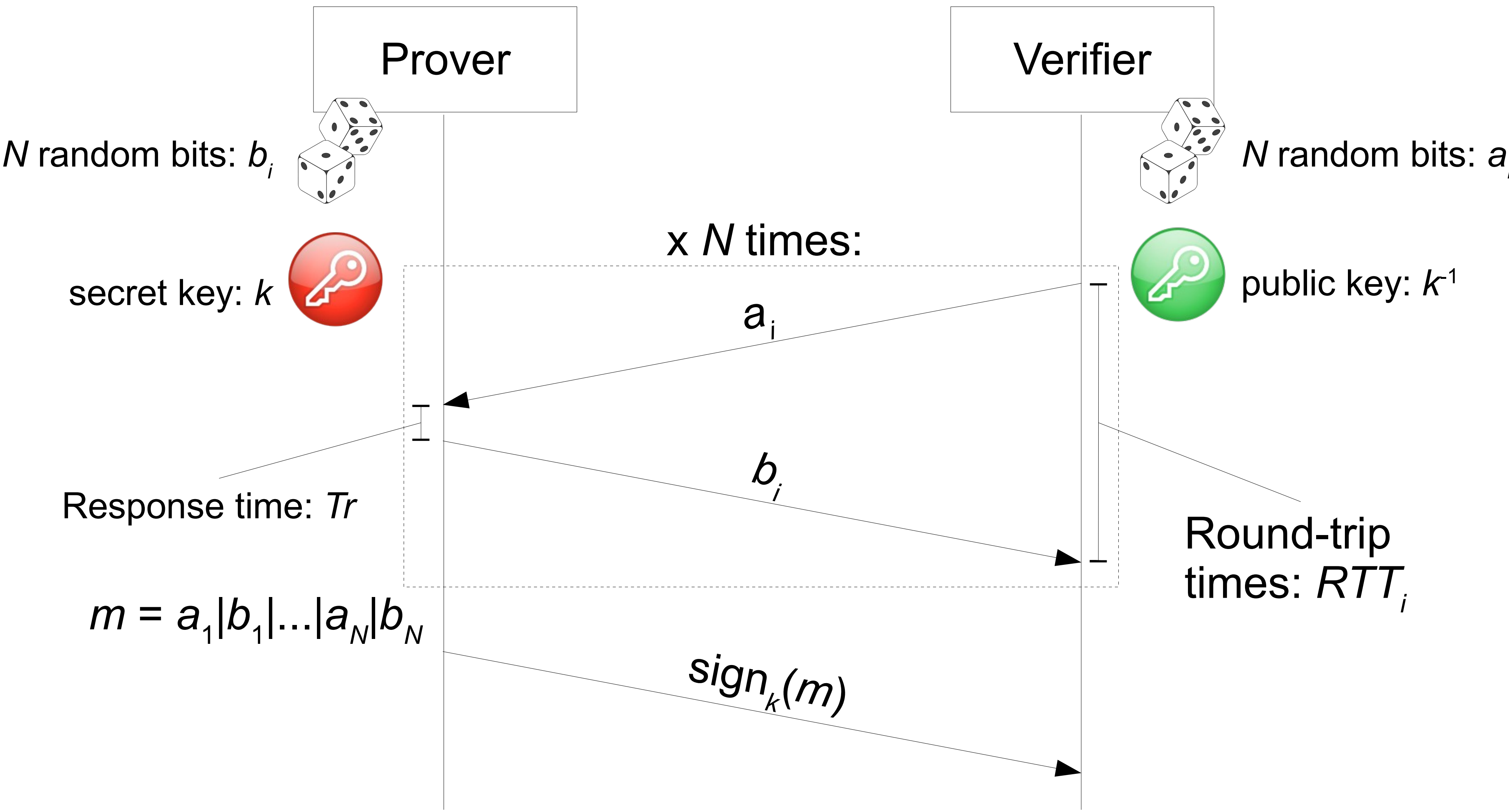
- From the routing point of view, the wormhole link is very convenient
- The adversary can split a wireless network in (roughly) two parts



Distance bounding protocol

- *Countermeasure:* precisely measure the round trip-time between a challenge and a response messages
- If the round-trip time is too large, reject the authentication (a mafia fraud could be present!)
- This is not enough!
- The adversary could build a relay link and actively anticipate the challenge and response messages
- The challenges and the responses must be *externally unpredictable*

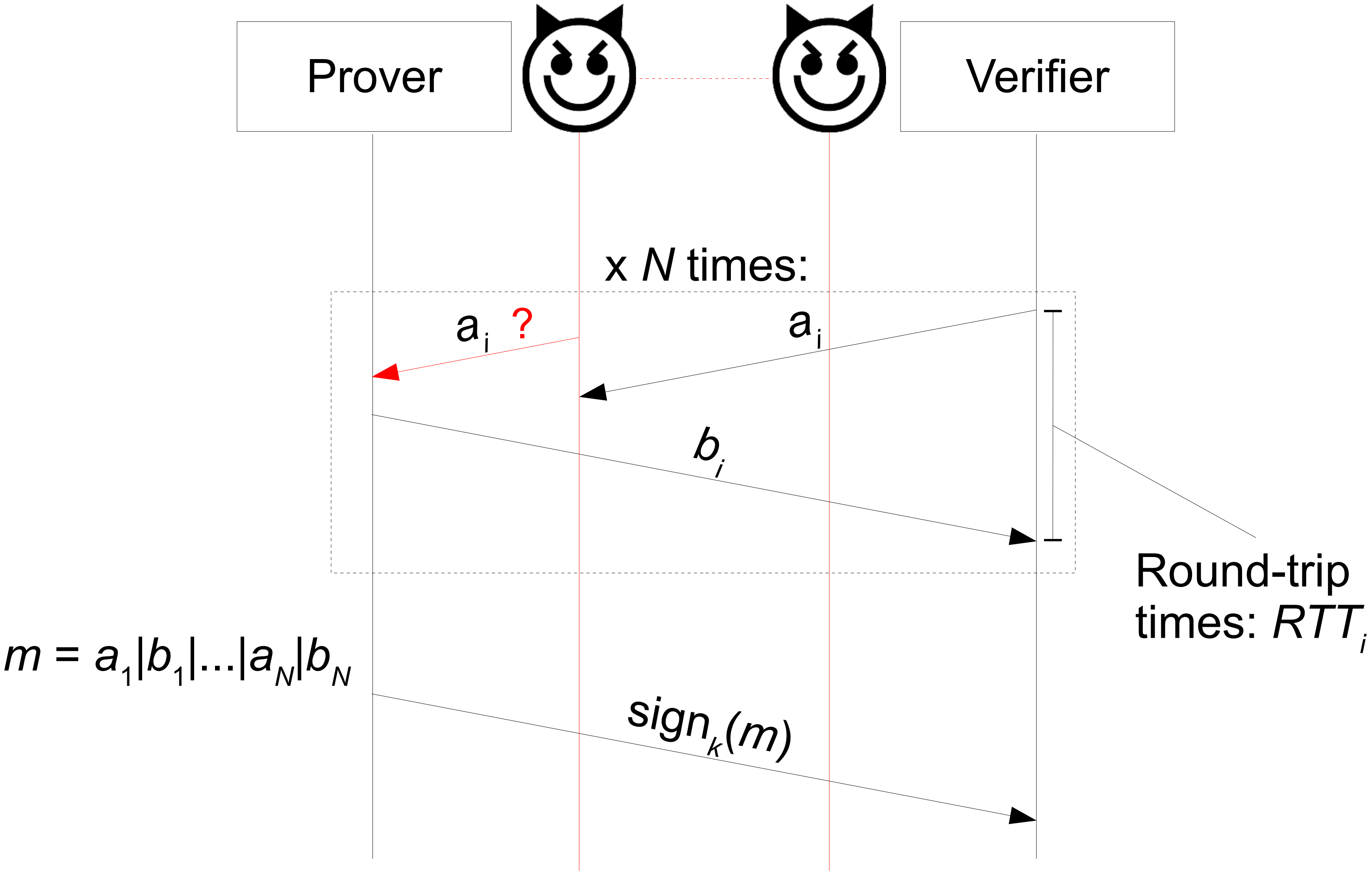
Brands-Chaum protocol (type I)



Brands-Chaum protocol (type I)

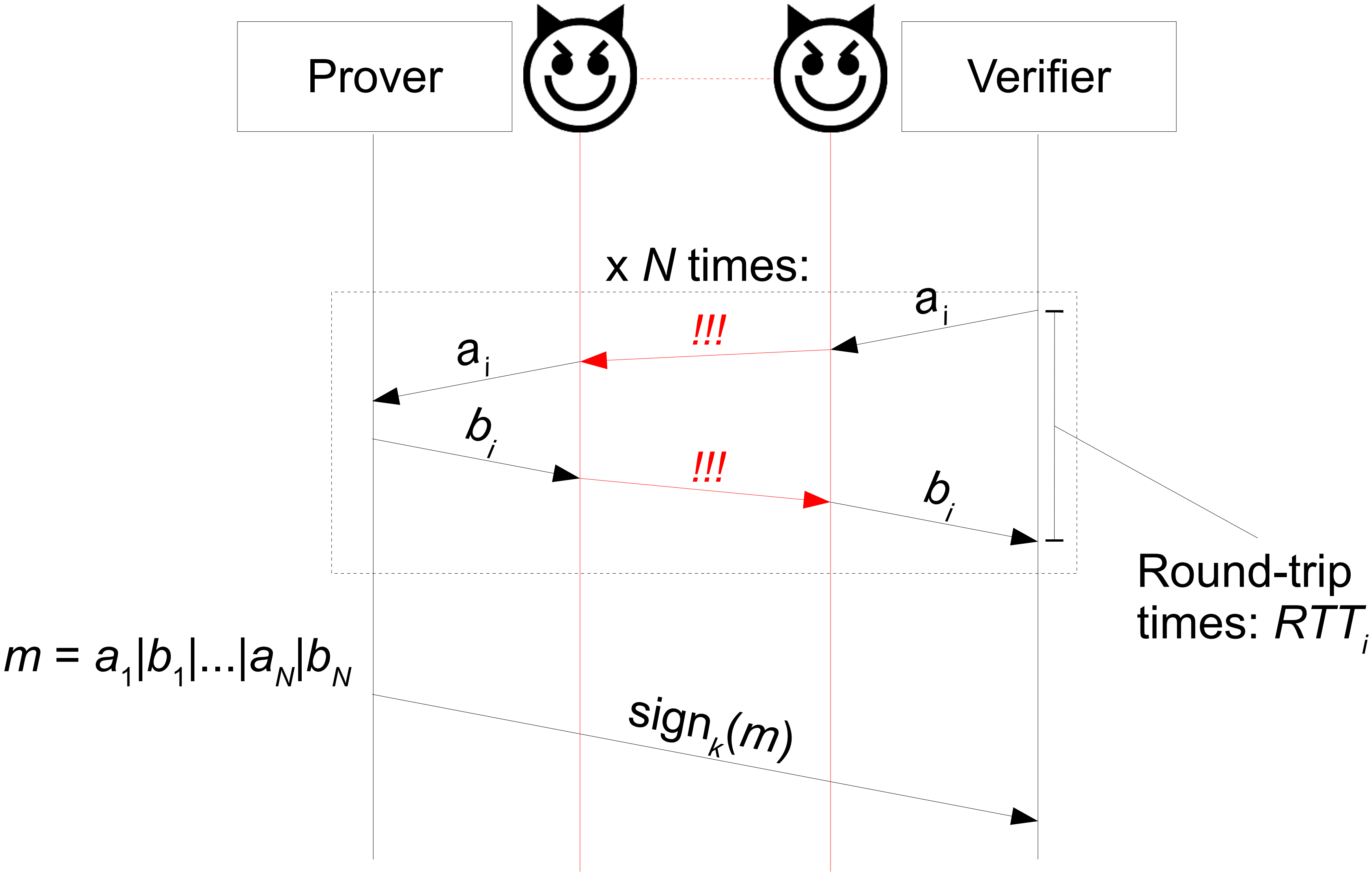
- Two general phases
  - **Rapid bit exchange (*real-time*)**: challenge and response bits are exchanged, the round-trip time is precisely measured
    - challenge and response bits are *externally unpredictable*
    - channel speed is *impassable* (typically radio or electrical, avoid sound!)
  - **Signature**: the prover signs the challenge and response bits with a secret
    - the device which sent the responses proves to be the prover

Brands-Chaum protocol (type I)



challenge and response bits are externally unpredictable

Brands-Chaum protocol (type I)



channel speed is impassable

Distance bounding protocol

- The verifier:
  1. Executes the protocol
  2. Verifies the validity of the signature
  3. Computes the *measured distance D* as:

     $D = \max(RTT_i) * c/2$

     c = speed of light
  4. Verifies that the measured distance is within a *proximity distance* D_{max}

     $D \le D_{max}$

Distance bounding protocol

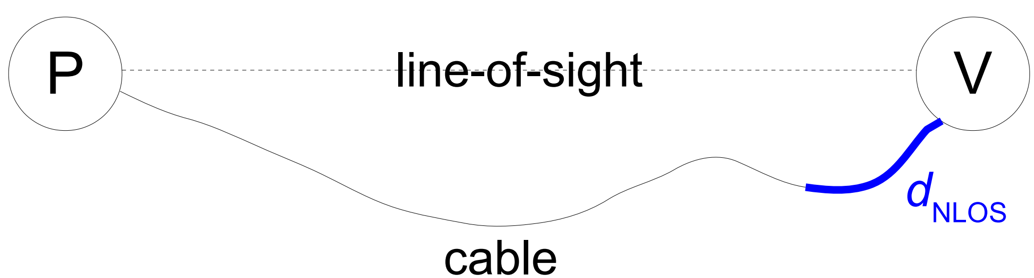
- The real distance *d* is given by:

  $d = \text{mean}(RTT_i - Tr) * v/2 - d_{NLOS}$

  Tr: response time

  $v < c$: real signal speed

  d_{NLOS}: component due to the non-line-of-sight path of the signal



Distance bounding protocol

- The measured distance is always longer than the real one:

  $D >= d$

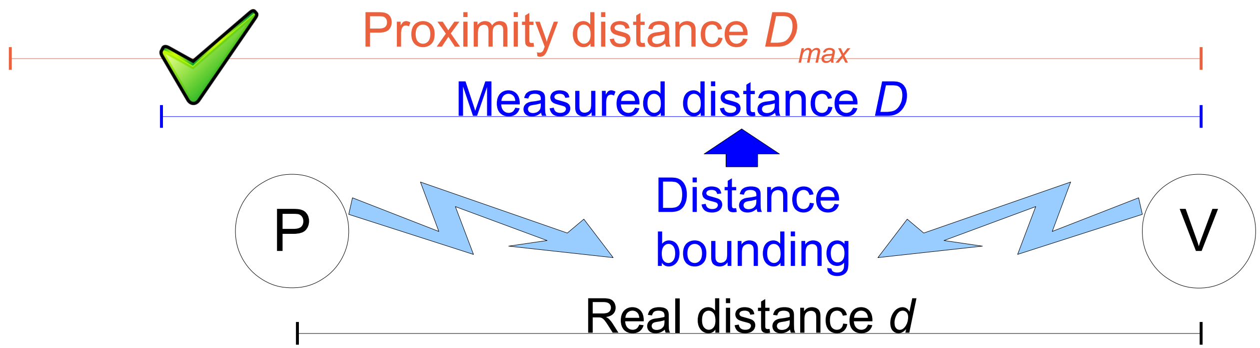
  $(\max(RTT_i)^*c/2) >= (\text{mean}(RTT_i - Tr)^*v/2 - d_{\text{NLOS}})$

- The term with the biggest impact is Tr
- If we design the prover to respond in $Tr >= Tr_{min}$ time, we can measure a more accurate distance (*accuracy improvement*):

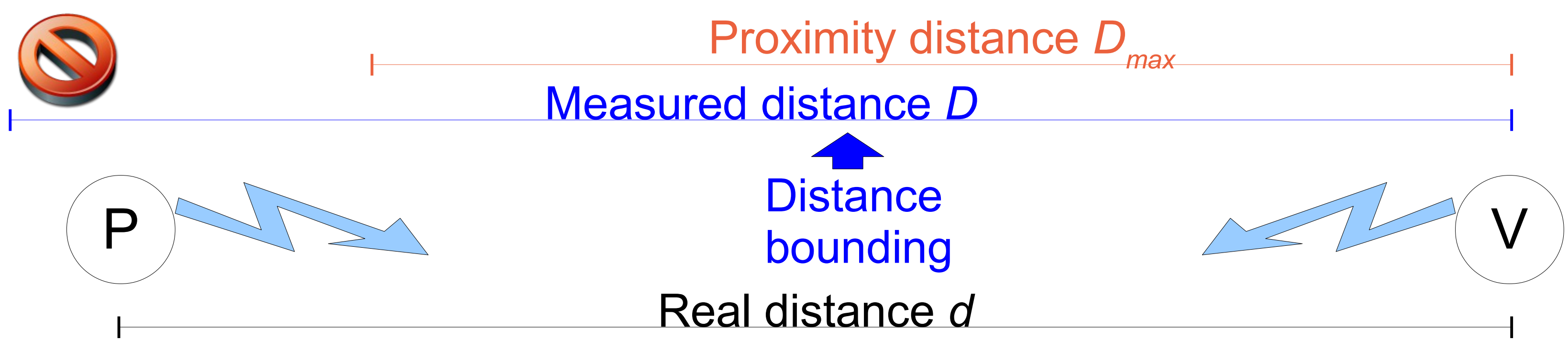
  $D = \max(RTT_i - Tr_{min})^*c/2$

Distance bounding protocol

- Honest case:



- Adversarial case:



Distance bounding protocol

- To mount a Mafia fraud, the adversary should build a *time-gaining relay*
- A time-gaining relay is a link that delivers *in advance* the challenge and/or the response bits
- However:
  - She cannot guess them in advance (unpredictability)
  - She cannot make them travel quicker than light (unpassability)

Distance bounding protocol

- What is the probability of successfully performing a time-gaining mafia fraud?
- The adversary has to anticipate *N* bit exchanges
- For each bit exchange, she has to guess and anticipate the response (or the challenge)

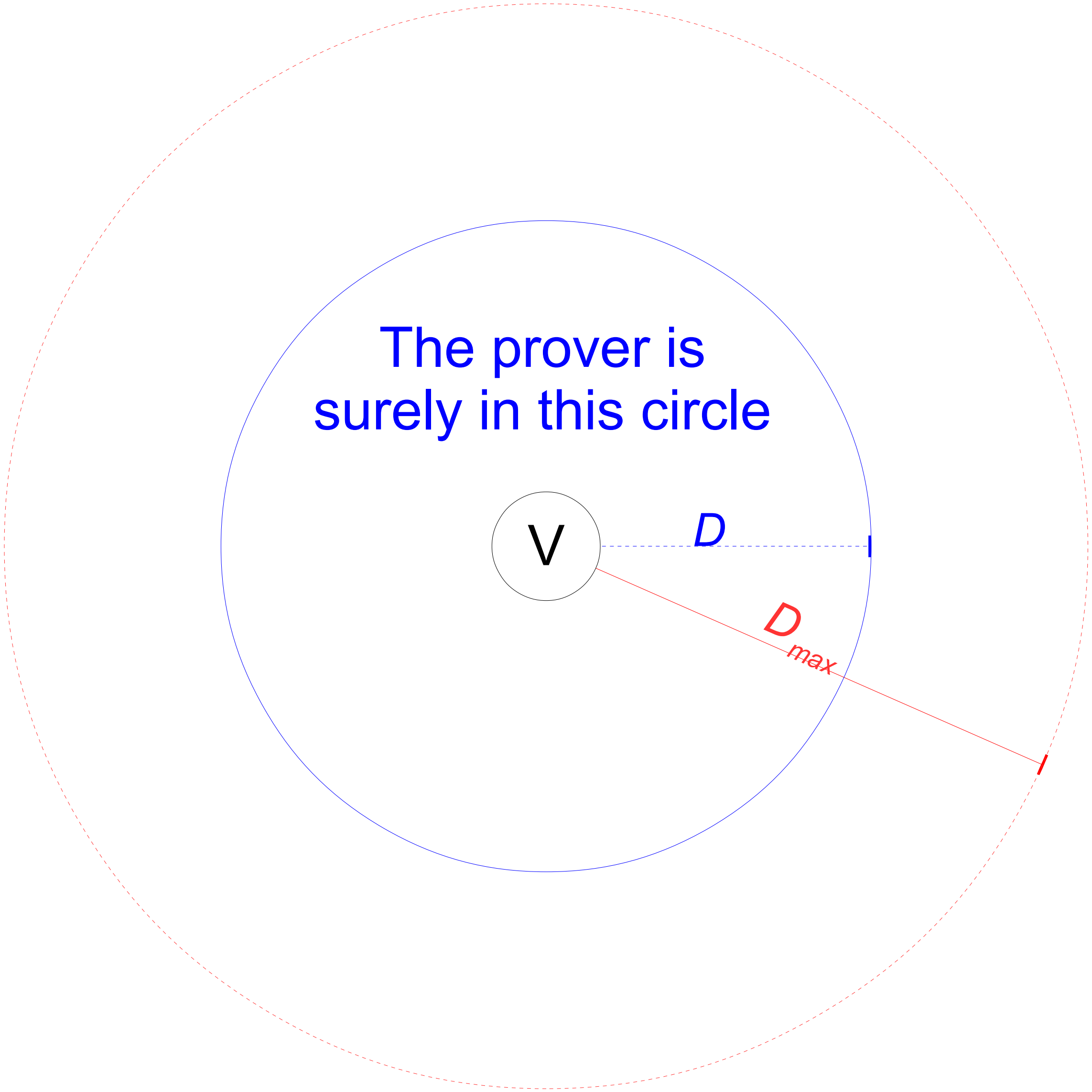
  $P_{1\text{-}round} = 1/2$

- Overall adversarial success probability:

  $P_{adv} = (1/2)^N$ (*negligible* with *N*)

  with *N*=128: $P_{adv} = 3*10^{-39}$

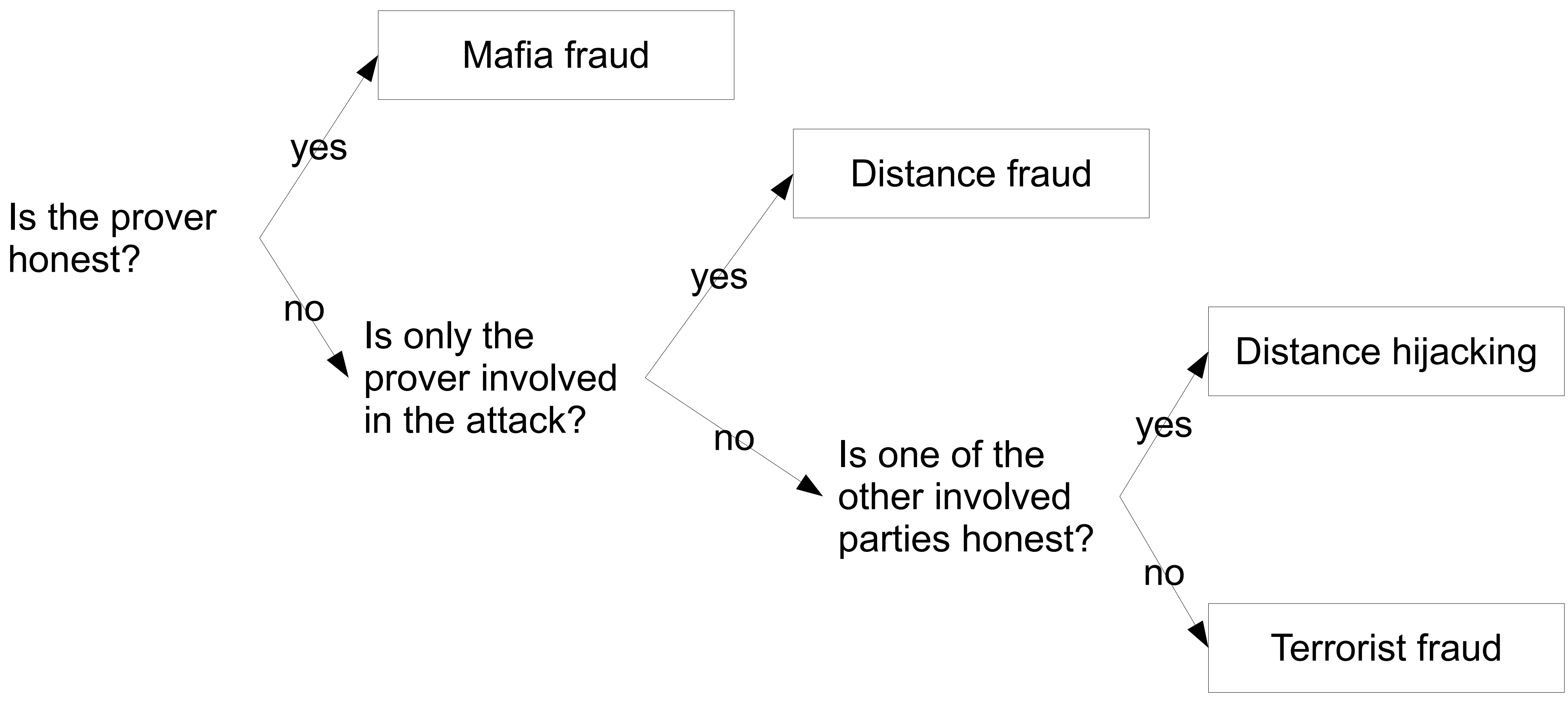
Distance bounding protocol



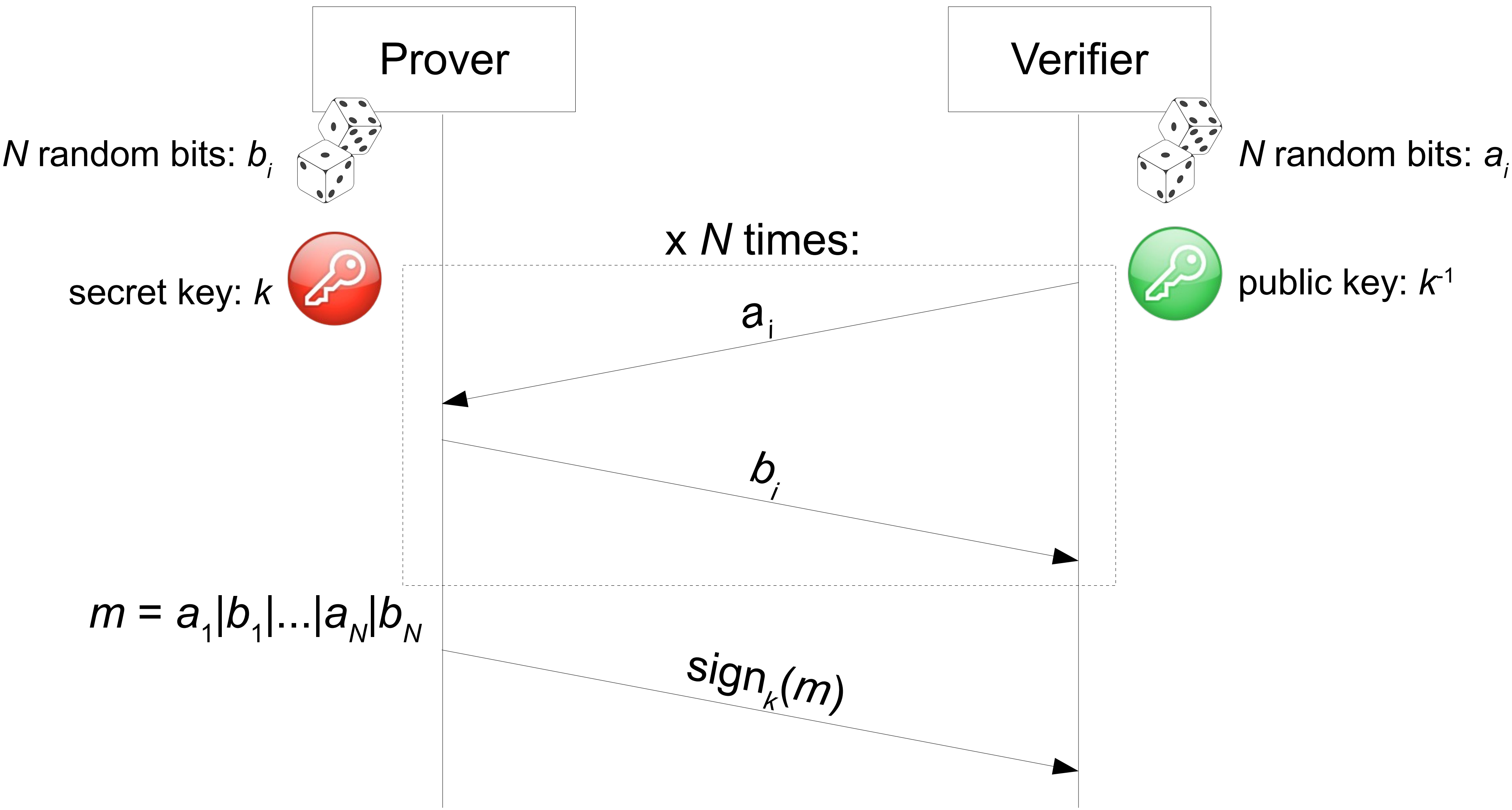
Other types of frauds

- *Mafia fraud*: an *external* adversary builds a relay link between P and V, and makes the distance appear shorter (*time-gaining relay*)
- *Distance fraud*: P itself is malicious, and makes its distance from V appear shorter (*time-gaining response*)
- *Terrorist fraud*: a malicious P colludes with an external adversary to make its distance from V appear shorter
- *Distance hijacking*: a malicious P leverages on another (honest) P to make its distance from V appear shorter

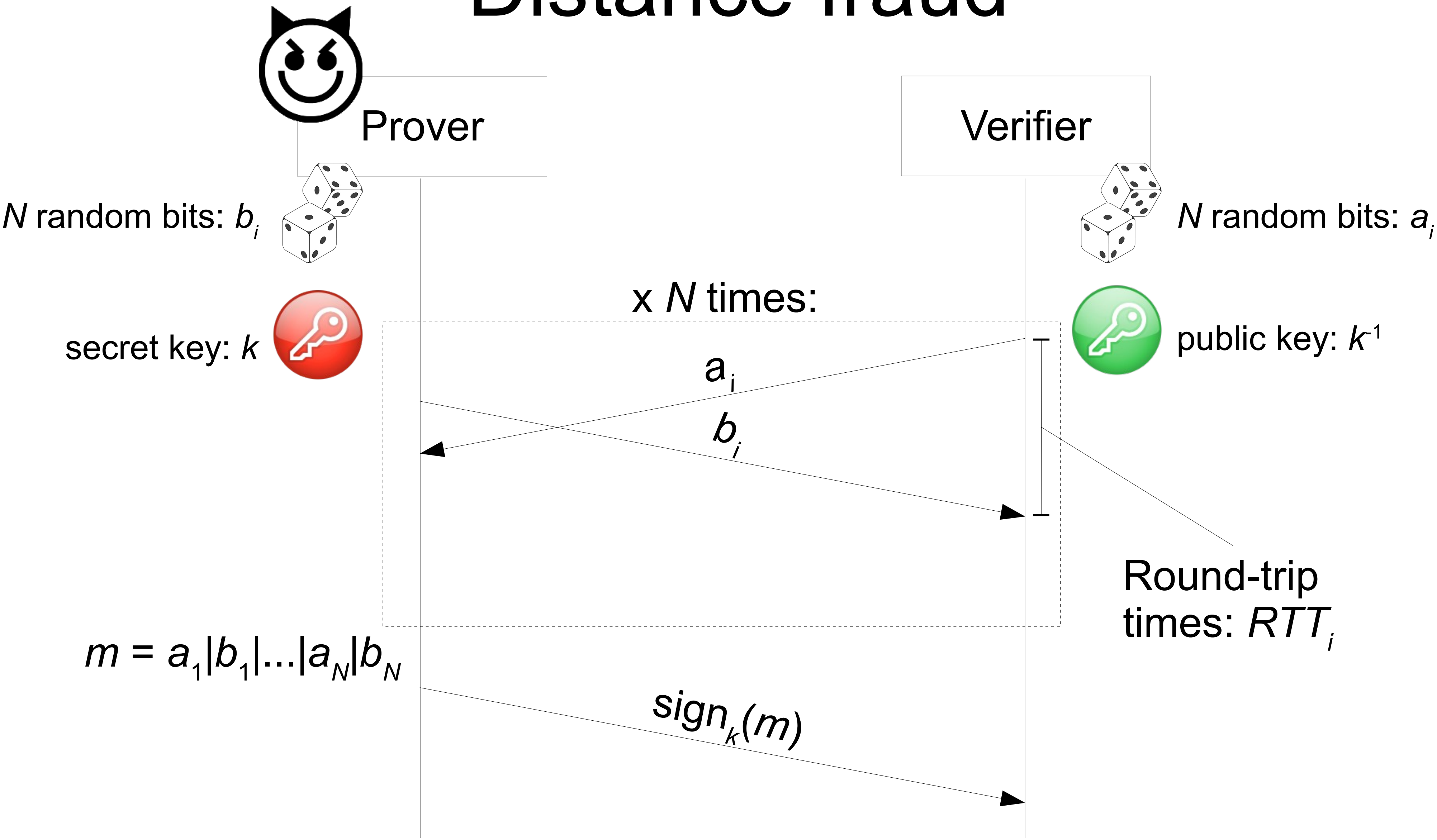
Other types of frauds



Distance fraud



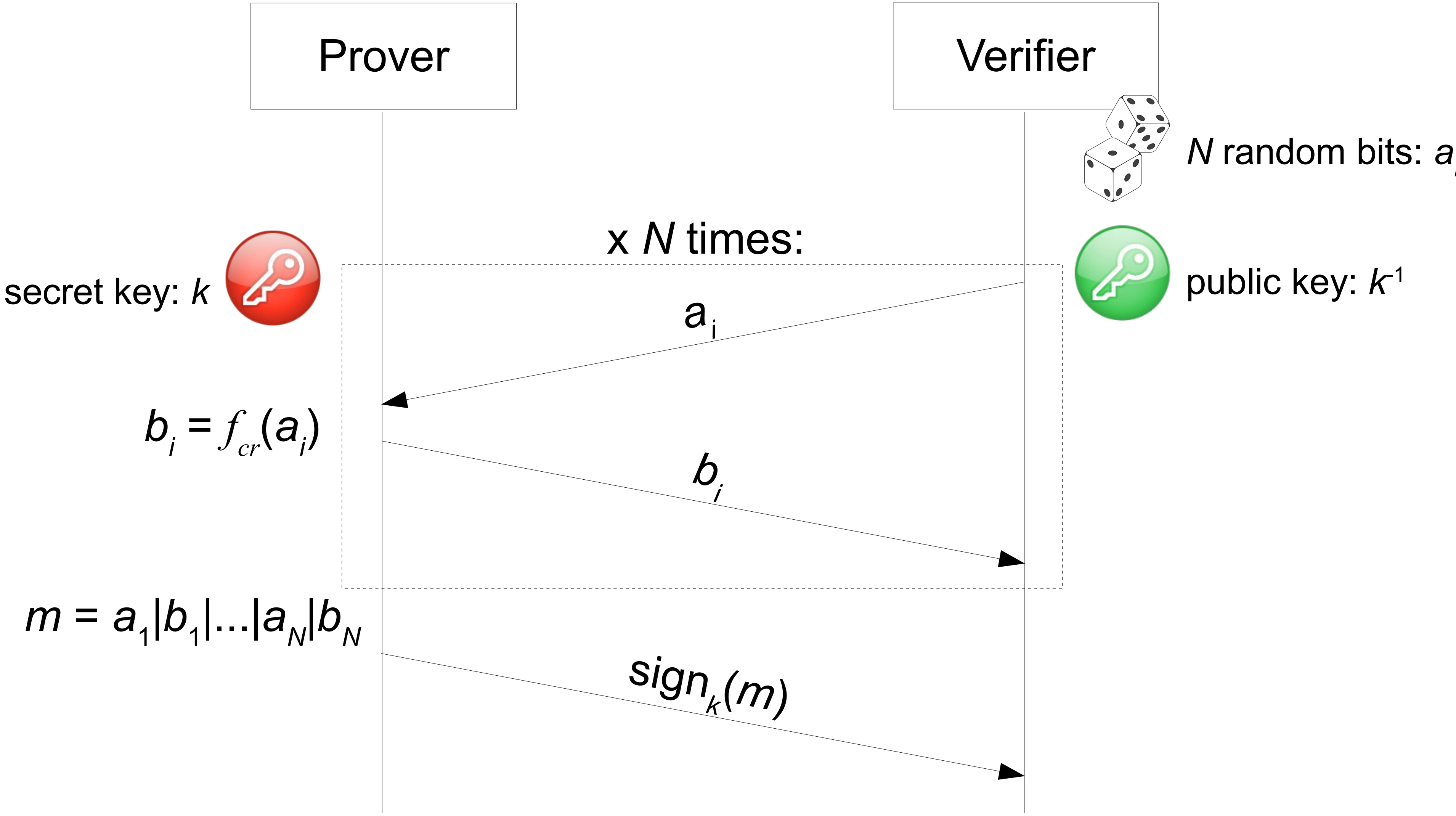
Distance fraud



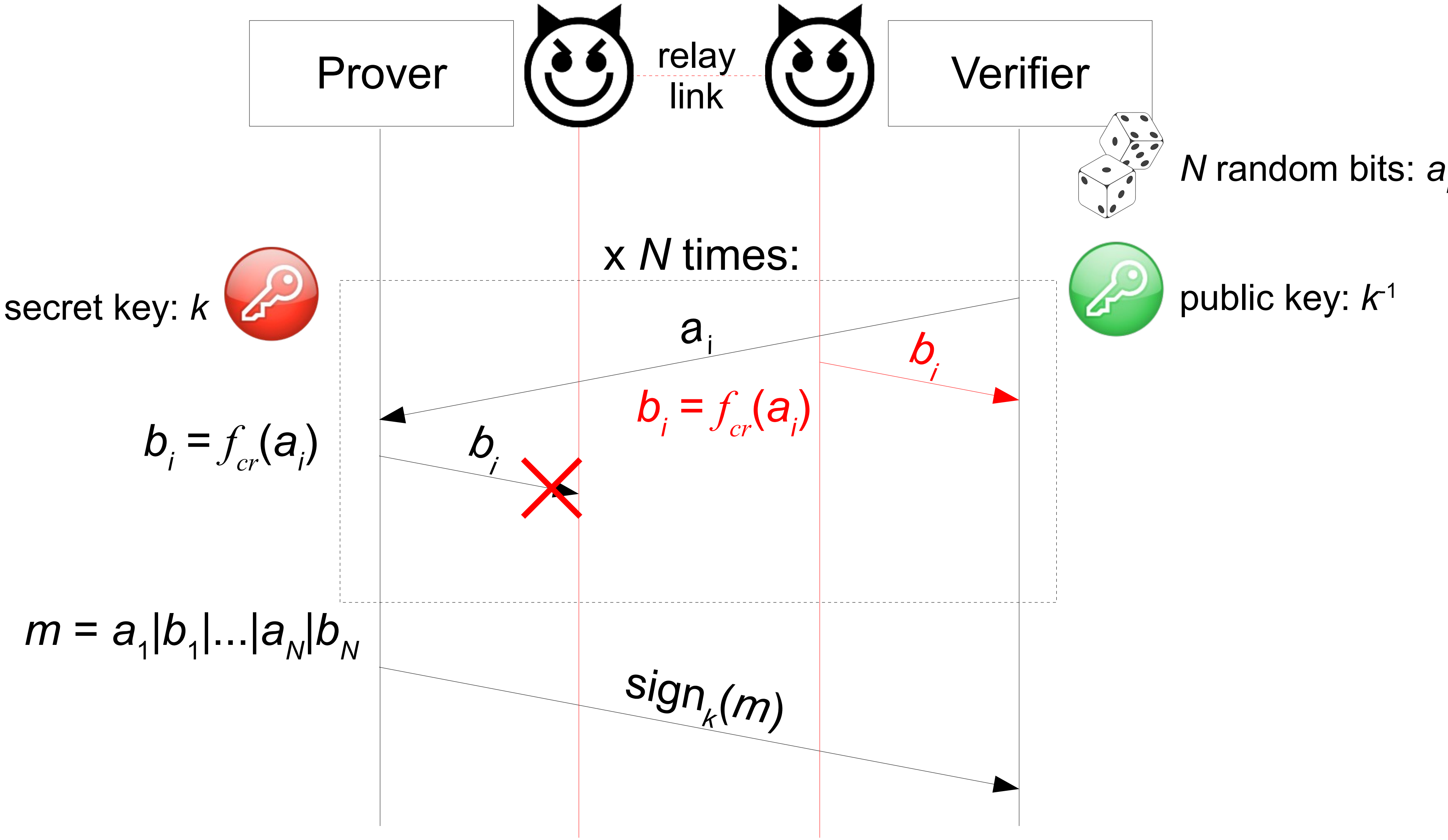
Distance fraud

- *Countermeasure:* the response bits must depend on the challenge bits
- *Challenge-response function*: $b_i = f_{cr}(a_i)$
- In this way, the response bits becomes *externally predictable* (mafia fraud vulnerability!)

Distance fraud



Distance fraud



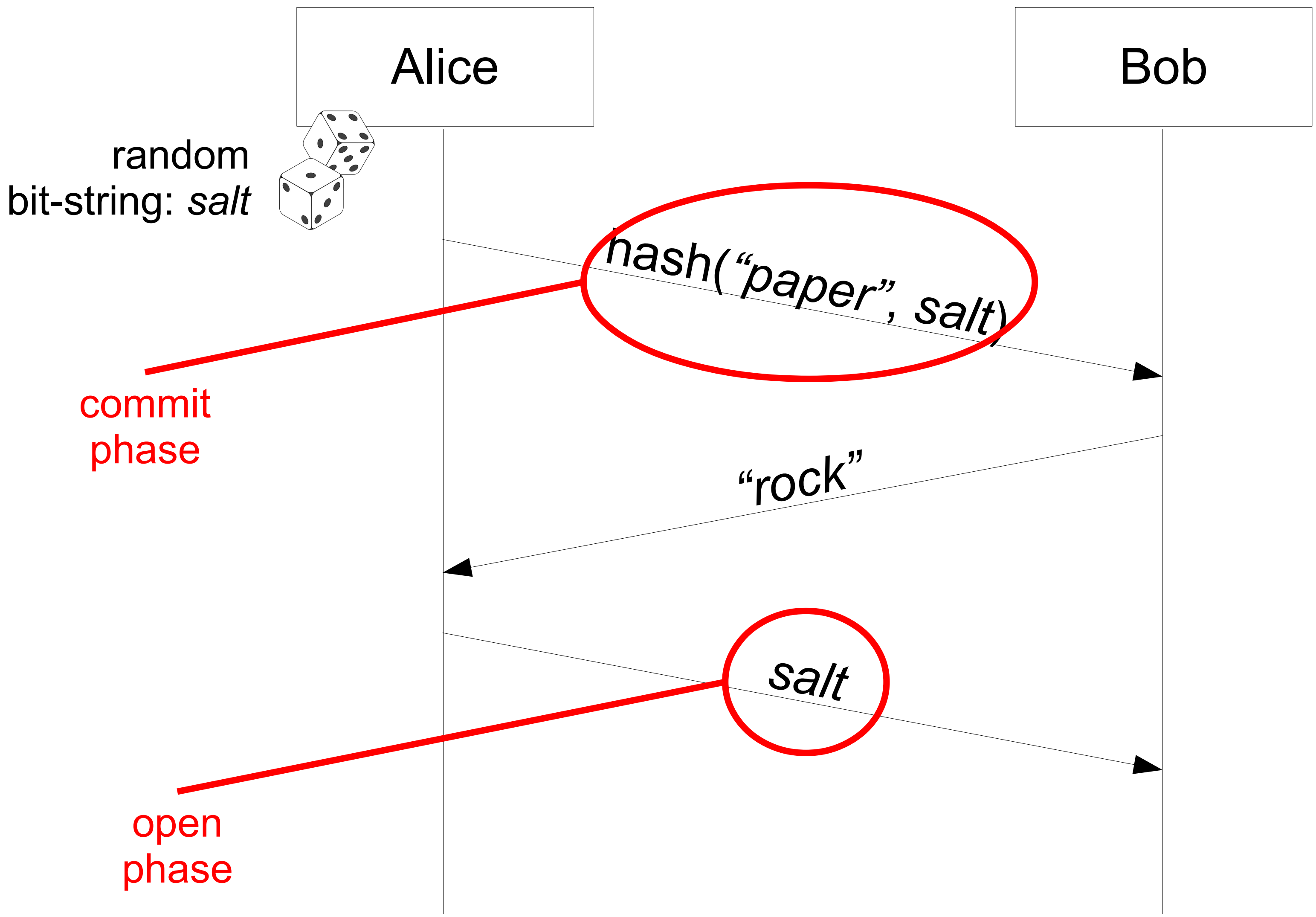
Commitment scheme

- Bob and Alice want to play *rock-paper-scissor* by email
- Suppose there is not a trusted third entity
- If Alice first sends to Bob her choice (for example “*paper*”), Bob can cheat by changing his choice on-the-fly (for example “*scissor*”)
- Who plays for second *always wins*

Commitment scheme

- A commitment scheme is a cryptographic protocol which allows a party to *commit to* a value without *revealing* it
- "To commit to a value" = to be forced to use a particular value afterward

Commitment scheme



Distance fraud

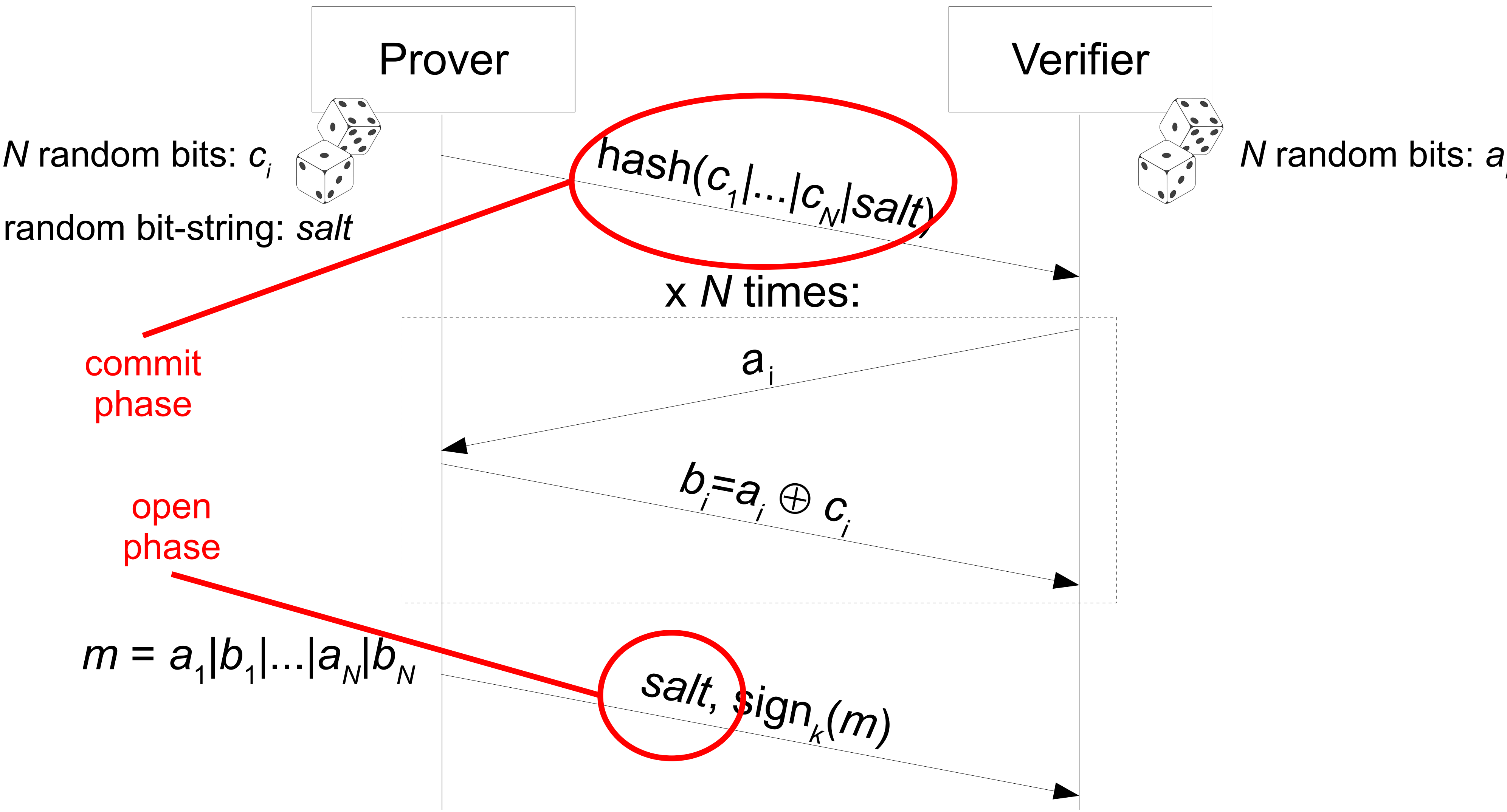
- *Challenge-response function:* $b_i = \cancel{f_{cr}(a_i)}$
- *Challenge-response function*: $b_i = f_{cr}(a_i, c_i)$
- The prover *commits* the bits c_i

Distance fraud

- The challenge-response function cannot be too complex, because the prover must respond timely

$$b_i = f_{cr}(a_i, c_i) = a_i \oplus c_i$$

Brands-Chaum protocol (type II)



It resists against mafia fraud and distance fraud

Brands-Chaum protocol (type II)

- Four general phases
  - **Commit**: the prover "promises" to use a particular sequence of bits c_i, without revealing it
  - **Rapid bit exchange (*real-time*)**: challenge and response bits are exchanged, the round-trip time is precisely measured
    - response bits must depend on the challenge bits and the committed bits
  - **Commit open**: the prover reveals the committed bits
  - **Signature**: the prover signs the challenge and response bits with a secret

Brands-Chaum protocol (type II)

- The verifier:
  1. Executes the protocol
  2. Verifies the validity of the commitment
  3. Verifies the validity of the signature
  4. Computes the *measured distance D* as:

     $D = \max(RTT_i) * c/2$

     c = speed of light
  5. Verifies that the measured distance is within a *proximity distance* D_{max}

     $D <= D_{max}$

Brands-Chaum protocol (type II)

- Can we make the accuracy improvement?

  $D = \max(RTT_i - \cancel{Tr_{min}})*c/2$

- No, because we cannot trust the prover to respect a minimal response time Tr_{min}
- A dishonest prover could have a more powerful hardware
  - compute quicker the challenge-response function
  - respond quicker

Brands-Chaum protocol (type II)

- What is the probability of successfully performing a time-gaining mafia fraud?

  $P_{adv} = (1/2)^N$

- What is the probability of successfully performing a distance fraud?
- For each bit exchange, the dishonest prover has to guess the response:

  $P_{adv} = (1/2)^N$

Overview

- A *distance bounding protocol* is a protocol that permits us to establish a *secure upper bound* (*D*) to the distance between a “prover” and a “verifier”

$$d_{\text{V-P}} <= D$$

- The basic idea is to precisely measure the *round-trip time* between two unpredictable messages (a challenge and a response)

Overview

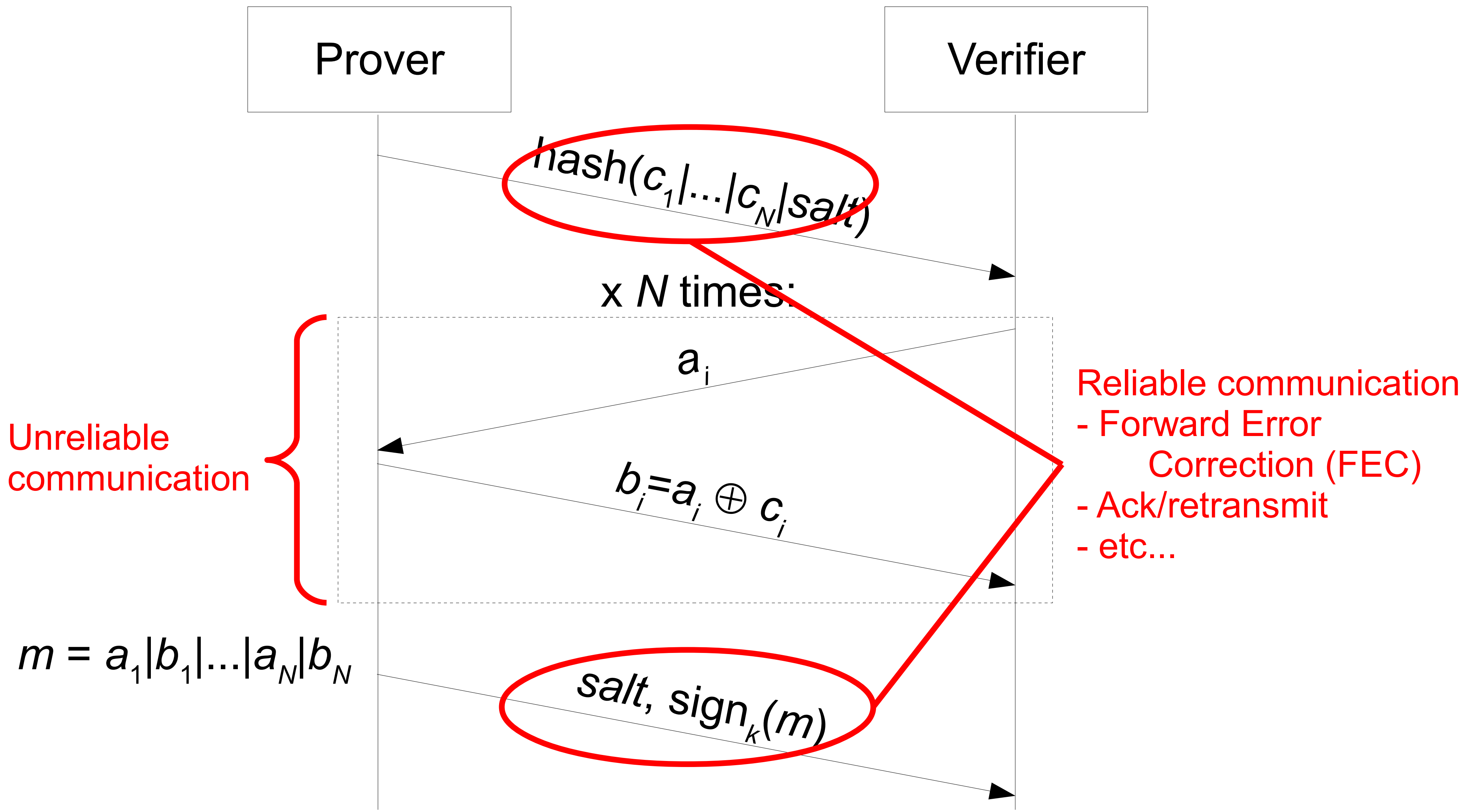
- *Brands-Chaum protocol (type I)* is a distance bounding protocol capable of resisting to *mafia fraud* (external adversary with relay link)
  - $P_{adv} = (1/2)^N$
- *Brands-Chaum protocol (type II)* is a distance bounding protocol capable of resisting to mafia fraud and *distance fraud* (dishonest prover that responds in advance)
  - Mafia fraud: $P_{adv} = (1/2)^N$
  - Distance fraud: $P_{adv} = (1/2)^N$

Distance bounding on RFID tags

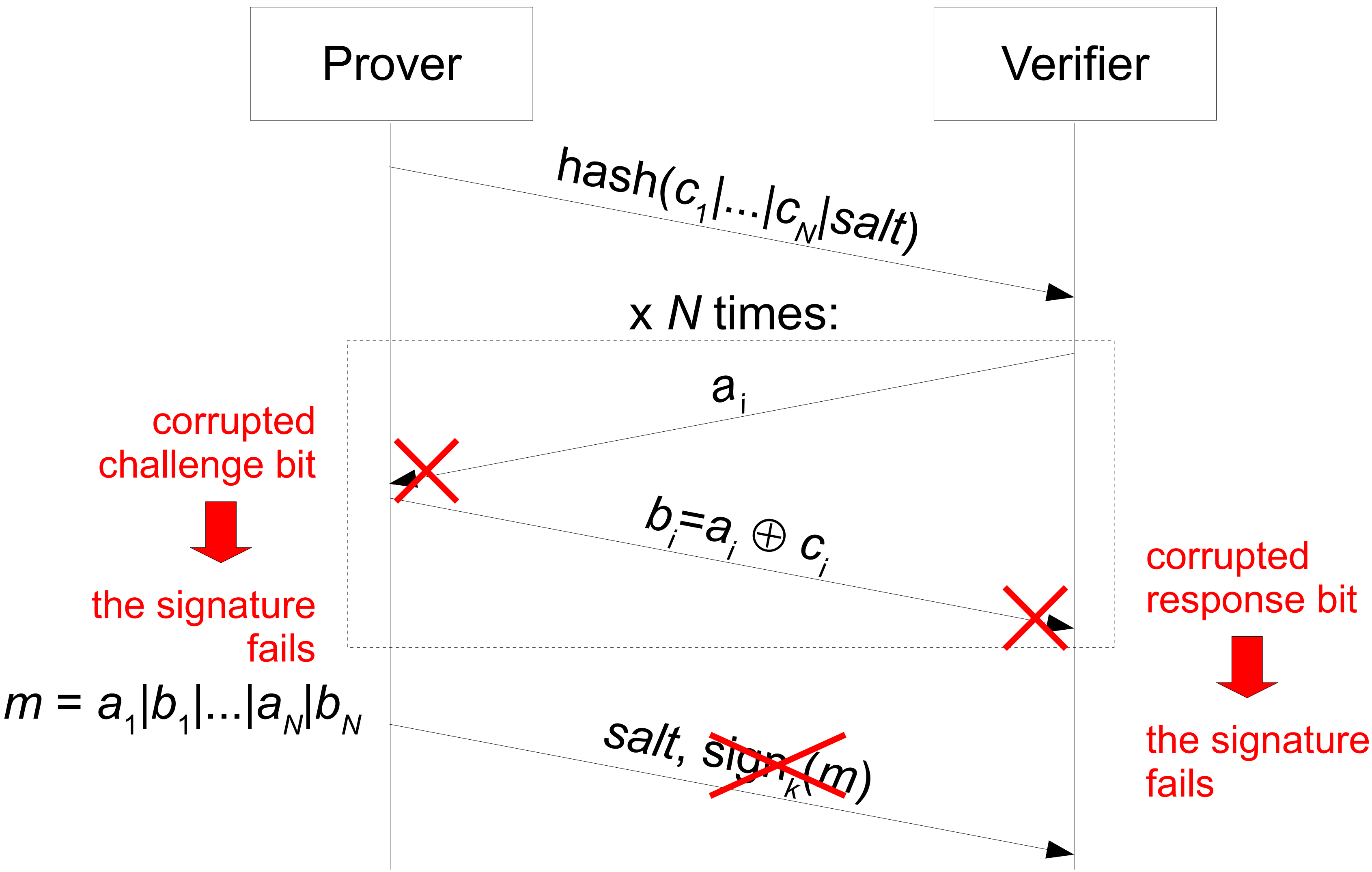
Distance bounding on RFID tags

- RFIDs are resource-constrained
  - It is expensive to equip them with unpredictable random number generators
- Wireless channels are *noisy*
  - The signature fails if one of the challenge-response bits gets corrupted
- RFIDs have an external (and *untrusted*) clock source
  - *Overclock attacks* are possible

Channel noise



Channel noise



Channel noise

- Probability of protocol failure:

  $P_{fail} = 1 – (1 – BER_{\text{V-P}})^N(1 – BER_{\text{P-V}})^N$

  $BER_{\text{V-P}}$, $BER_{\text{P-V}}$: bit error rates of P-V and V-P channels

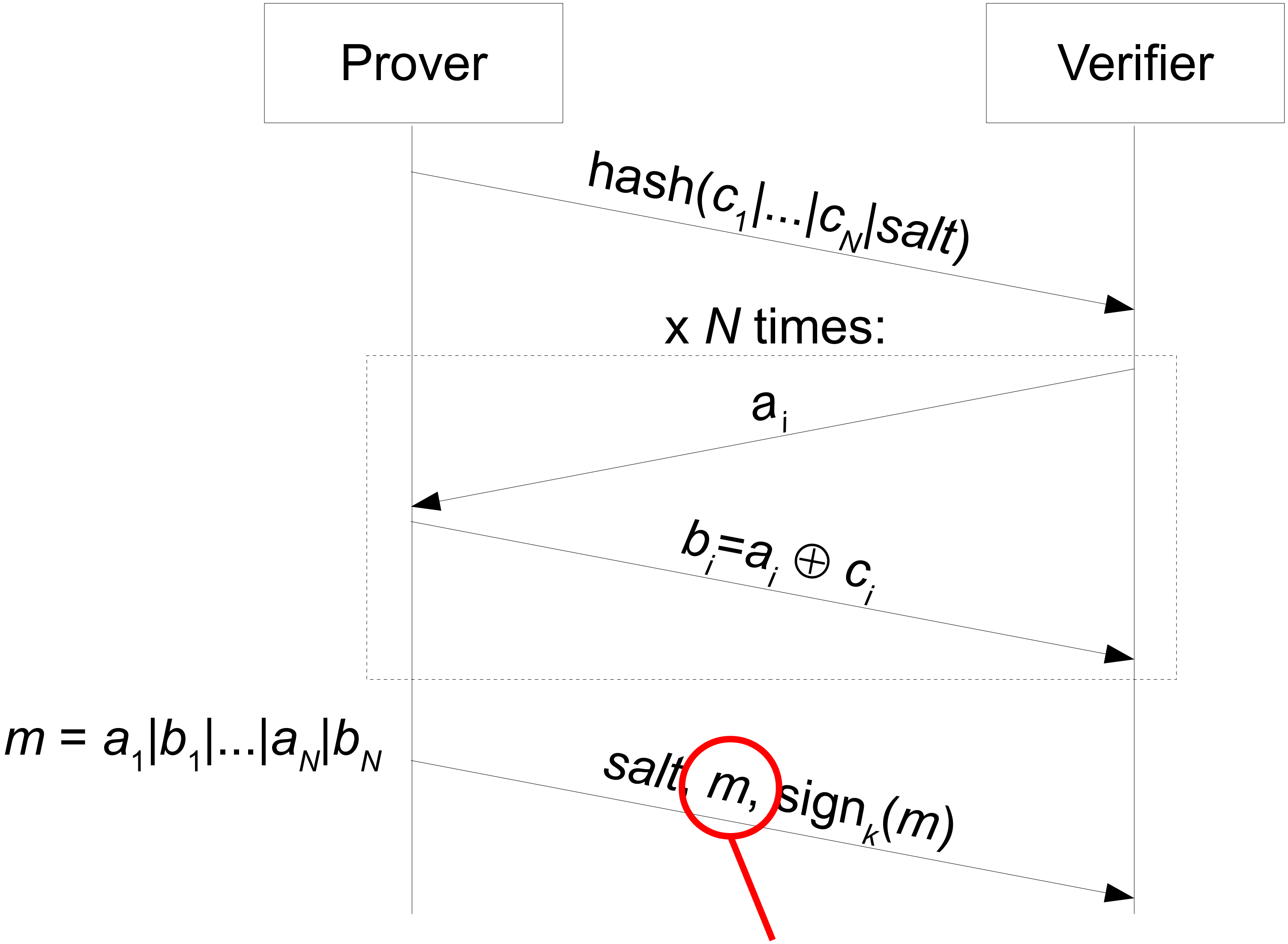
- With $BER_{\text{V-P}} = BER_{\text{P-V}} = 10^{-3}$ and N=128:

  P_{fail} = 23%!

Channel noise

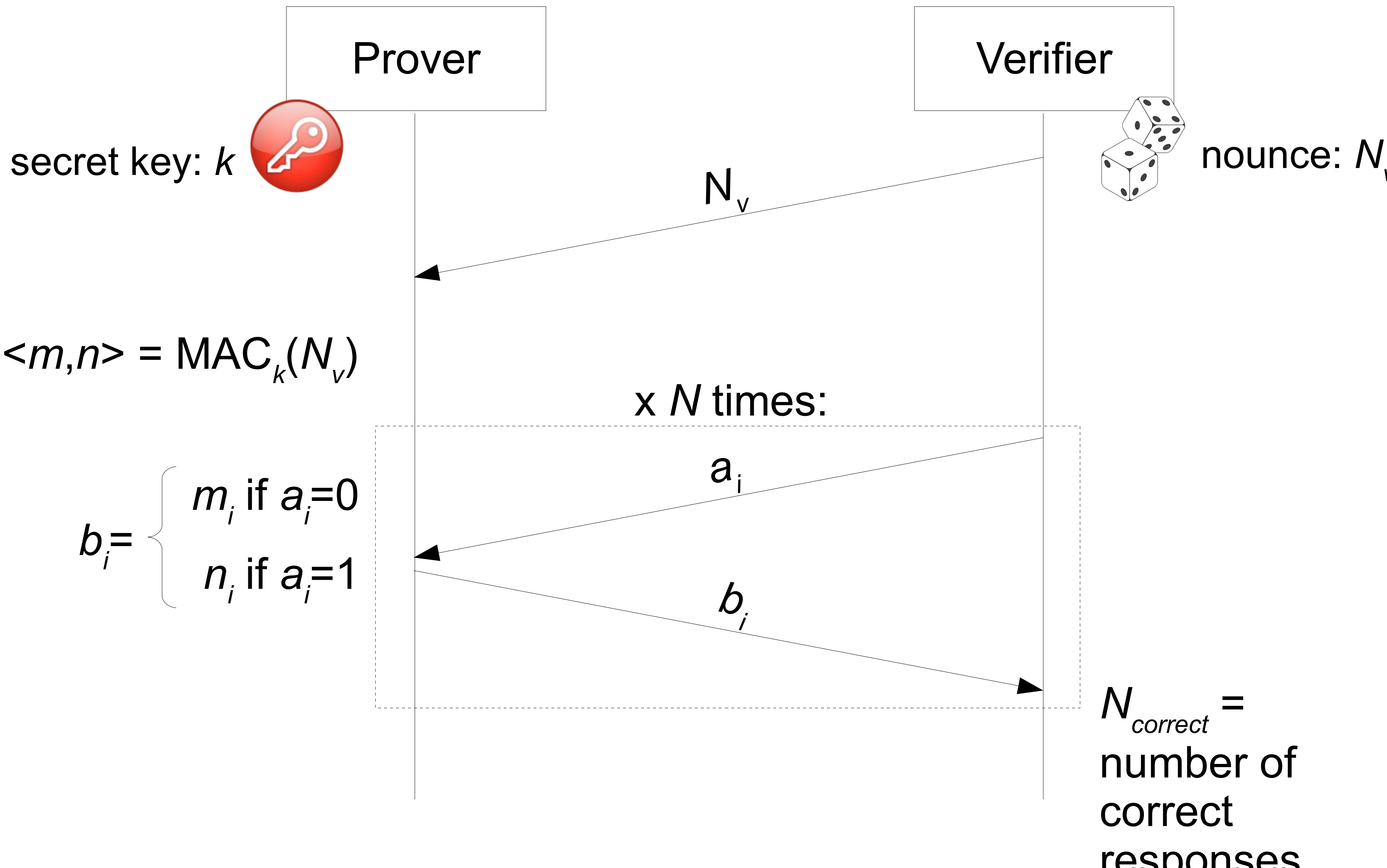
- Performing a *reliable* rapid bit exchange (for example with CRC) is burdensome:
  - More than one bit for every challenge and for every response
- … and insecure:
  - The dishonest prover could ignore the challenge's CRC and anticipate the response

Channel noise



Send again m with a reliable channel
The verifier checks how many corrupted bits

Hancke-Kuhn protocol



It resists against mafia fraud and distance fraud

Hancke-Kuhn protocol

- The verifier counts the number of correct responses $N_{correct}$
- If the correct responses are >= a threshold N_{accept}, the authentication is accepted

Noise tolerance

- Probability of protocol failure:

$$P_{fail} = \sum_{i=0}^{N_{accept}-1} \binom{N}{i} (1-\epsilon)^i \epsilon^{n-i}$$

where "ε" is the probability of receiving a corrupted response

$$\epsilon = \frac{BER_{\text{P-V}} + \left(1 - \left(1 - BER_{\text{V-P}}\right)\left(1 - BER_{\text{P-V}}\right)\right)}{2}$$

- With $BER_{\text{V-P}} = BER_{\text{P-V}} = 10^{-3}$, N=128, and N_{accept}=124:

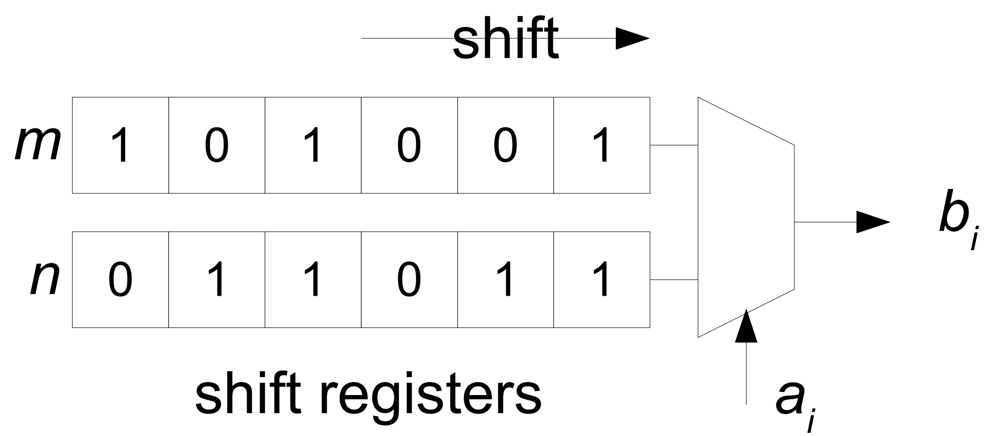
$P_{fail} = 2*10^{-6}$ (Brands-Chaum was P_{fail}=23%)

Hancke-Kuhn protocol

- Challenge-response function implemented with *shift registers*

externally unpredictable

$$b_i = f_{cr}(a_i, m, n)$$



Hancke-Kuhn protocol

- Two general phases
  - **Secret initialization**: prover and verifier agree to an *externally unpredictable secret* (*m*, *n*)
  - **Rapid bit exchange + signature (*real-time*)**: challenge and response bits are exchanged
    - The signature is contextual with the rapid bit exchange

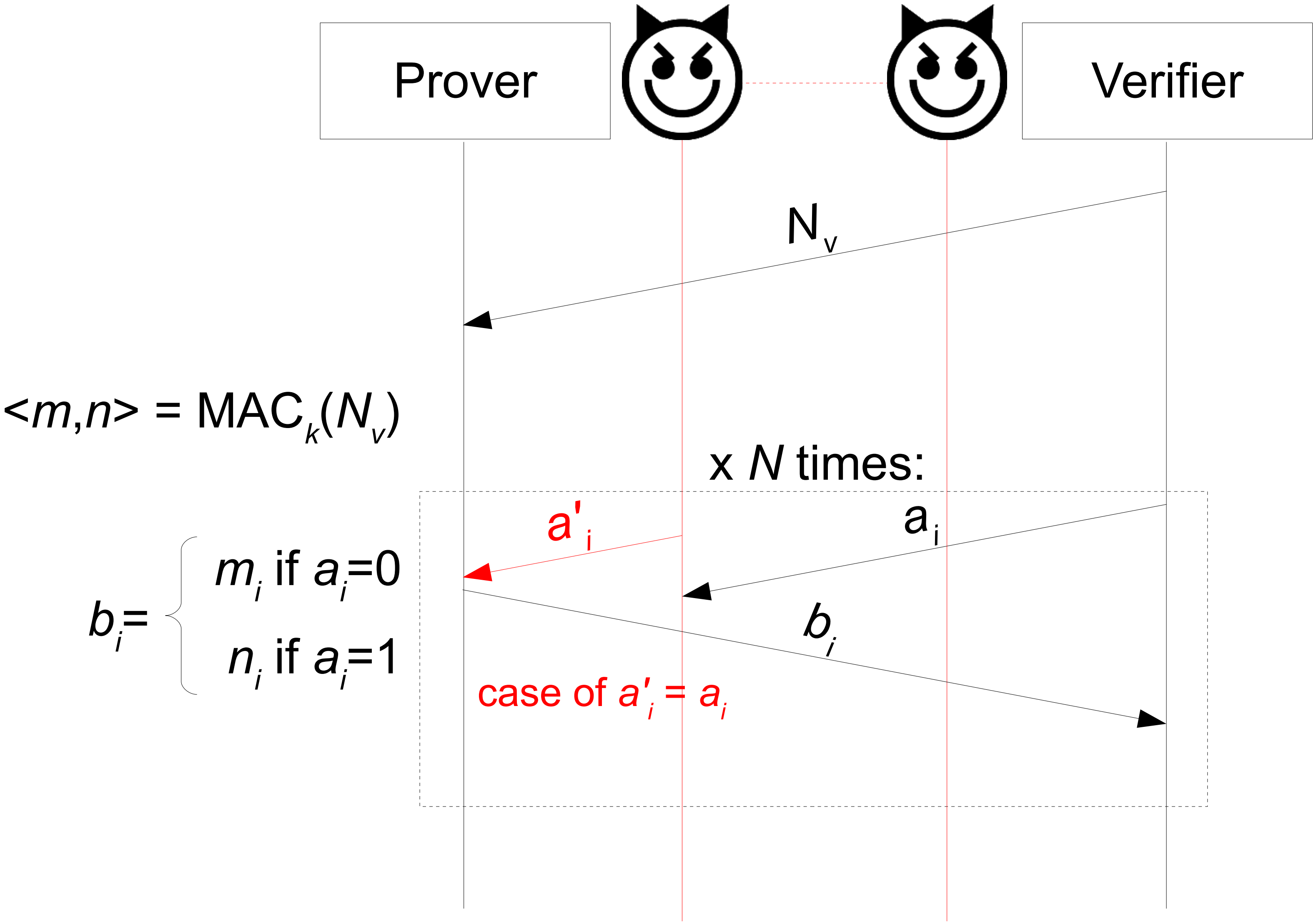
Hancke-Kuhn protocol

- The prover is not required to produce (and commit to) an unpredictable quantity
- No final signature
  - In practice, the *response bits* are the signature
- The overall quantity of messages is decreased (time efficiency)
- It is possible to tolerate a certain number of wrong response bits, due to channel noise

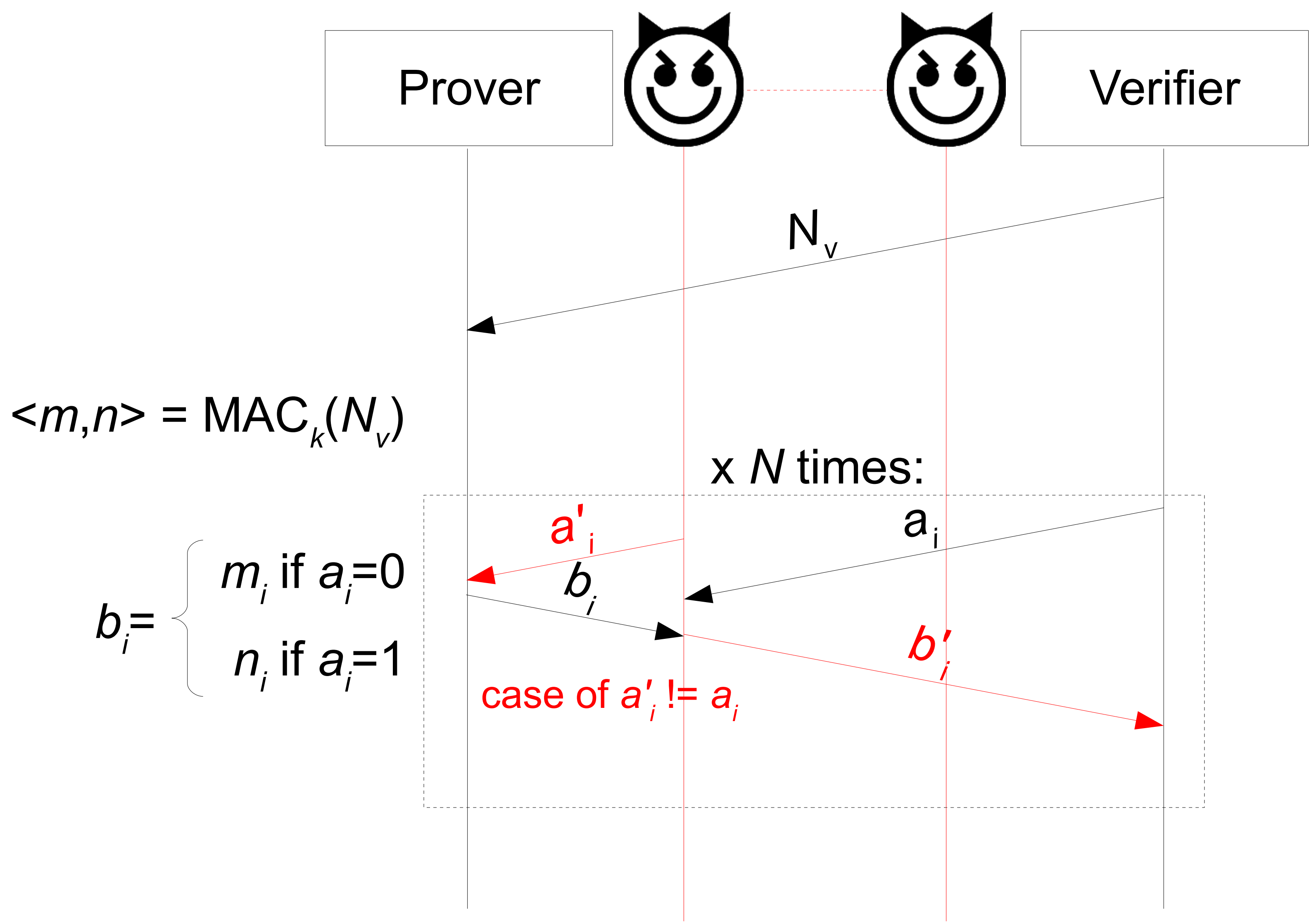
Double-chance guessing attack

- Hancke-Kuhn distance bounding is vulnerable to *double-chance guessing*!
  - the adversary tries to guess the challenge bit
  - if she fails, she has *another chance* by trying to guess the response bit

Double-chance guessing attack



Double-chance guessing attack



Double-chance guessing attack

- What is the probability of successfully performing the double-chance guessing attack?
- For each bit exchange, she has to perform double-chance guessing

  $P_{1\text{-}round}$ = 1/2 + 1/2*(1/2) = 3/4

- Overall adversarial success probability:

$$P_{adv} = \sum_{i=N_{accept}}^{N} \binom{N}{i} (3/4)^i (1/4)^{N-i}$$

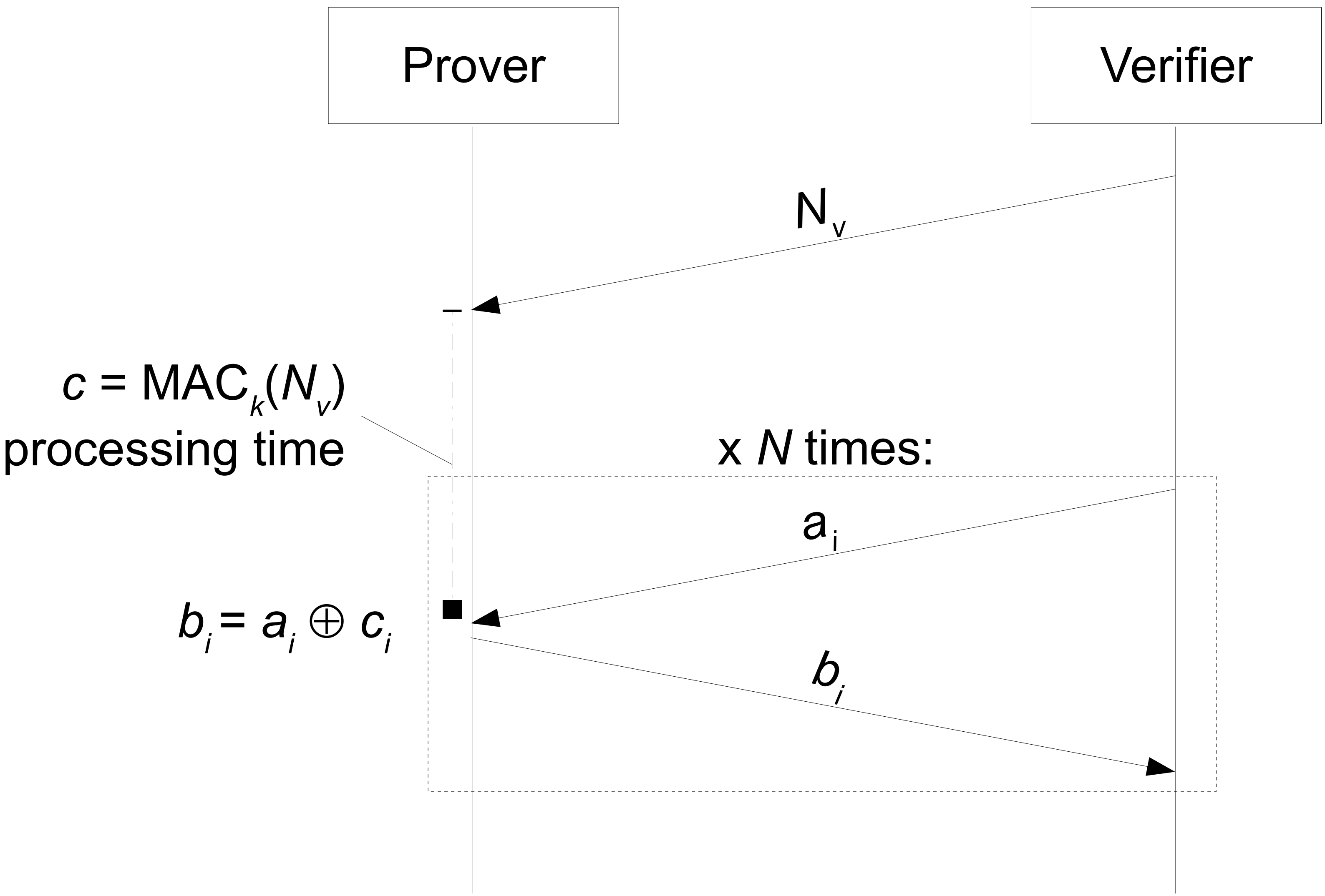
with N=128 and N_{accept}=124: $P_{adv} = 10^{-12}$

(Brands-Chaum was $P_{adv} = 3*10^{-39}$)

Overclocking attack

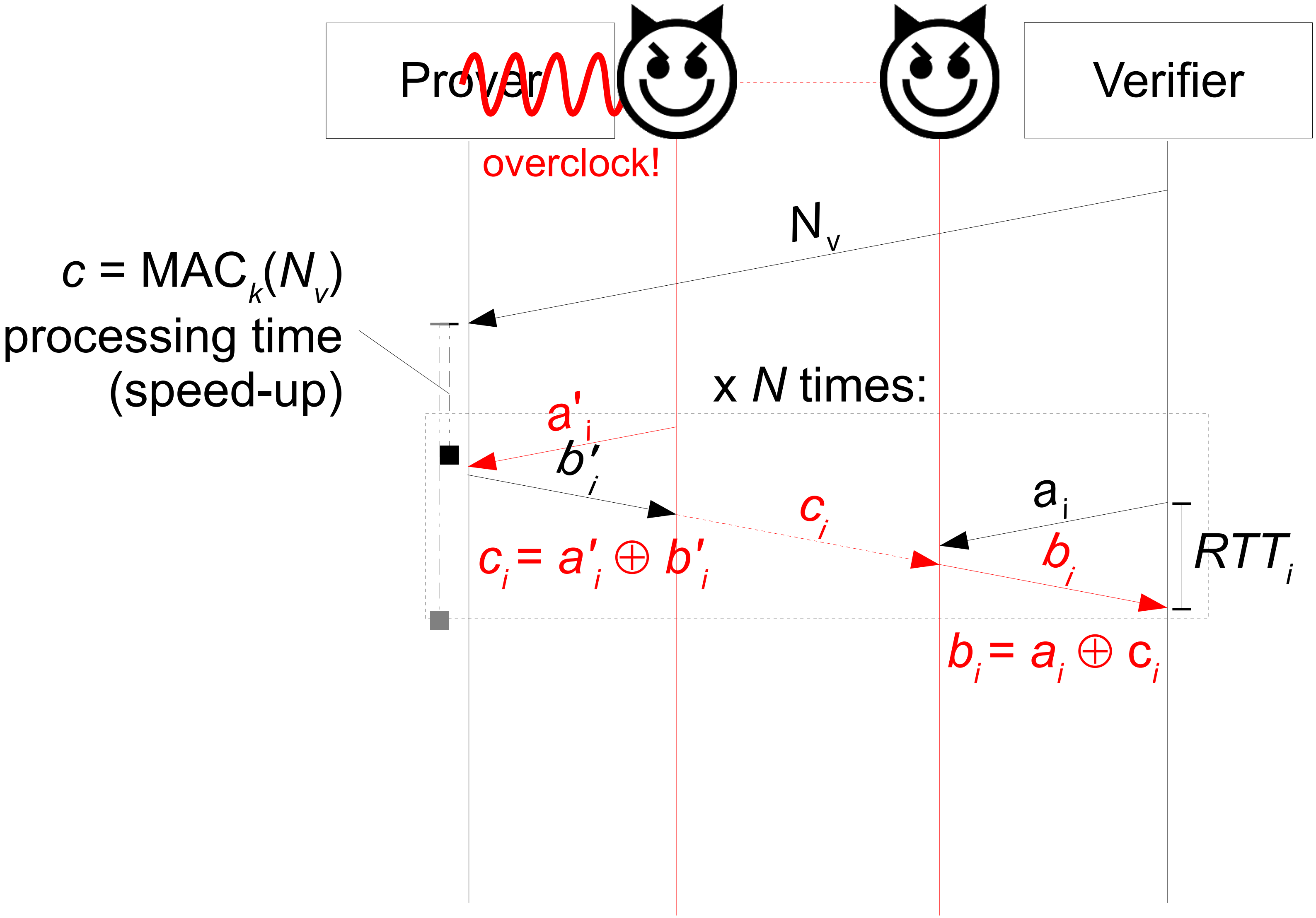
- RFIDs do not have an internal clock source
- Their clock source is untrusted
- An external adversary could overclock a legitimate prover to get the responses in advance
- *Countermeasure*: shift registers challenge-response function

Overclocking attack

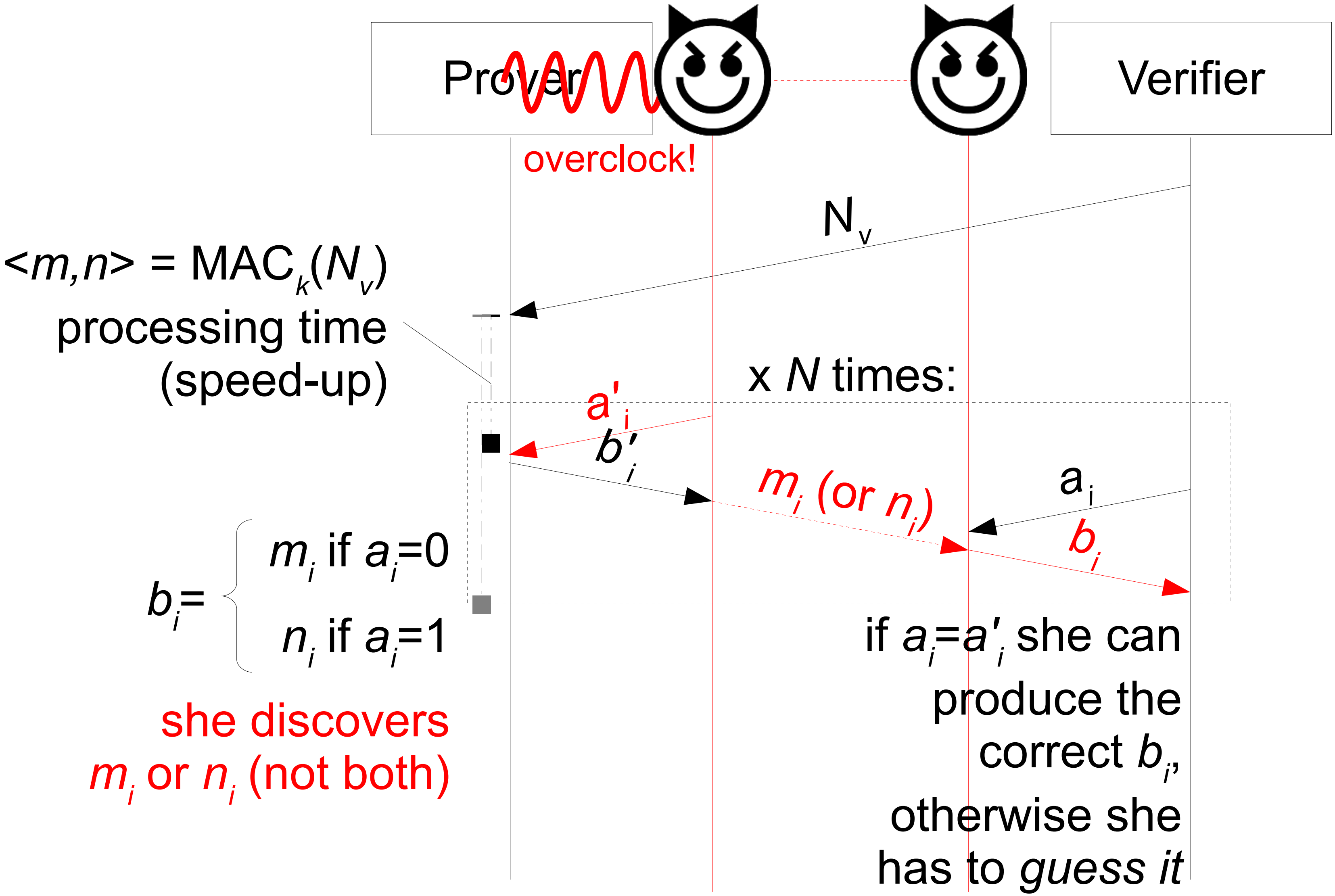


XOR-based version of Hancke-Kuhn protocol
is vulnerable to overclocking attack

Overclocking attack



Overclocking attack



Overclocking attack

- For each bit exchange:
  - the adversary discovers a register bit, and hopes that it is the "useful" one (P=1/2)
  - if she fails, she tries to guess the response bit (P=1/2)

  $P_{1\text{-}round} = 3/4$

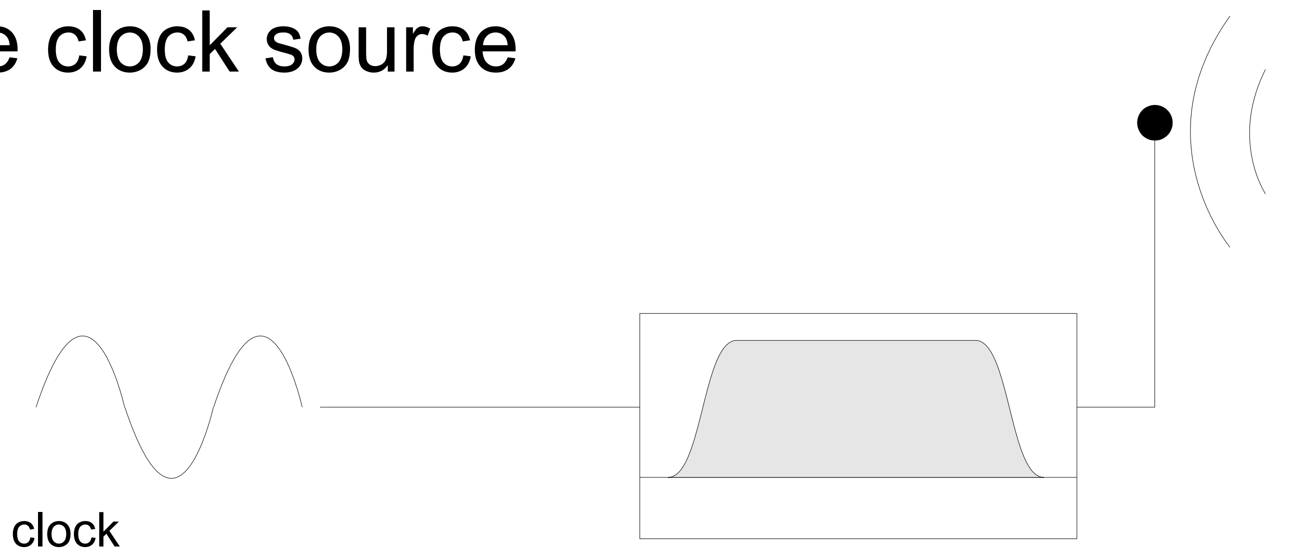
- Overall adversarial success probability:

$$P_{adv} = \sum_{i=N_{accept}}^{N} \binom{N}{i} (3/4)^i (1/4)^{N-i}$$

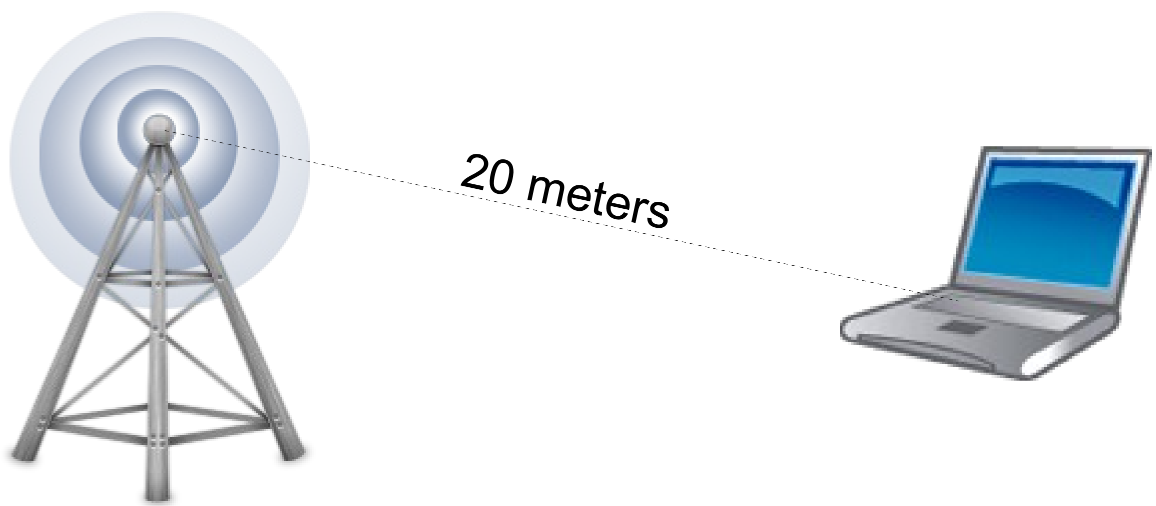
(same as double-chance guessing attack)

Overclocking attack

- To be sure of being successful, the adversary should perform *twice* all the *N* bit exchanges with the legitimate prover, and discover both *m* and *n* registers
- She should perform a *huge overclock*, which easily avoidable by means of a *low-pass filter* on the clock source



Frame-based distance bounding



Frame-based distance bounding

- Prover and verifier are far away (up to 20-30m)
- Every message comes with a *PHY header* (for time synchronization and demodulation infos)
- The PHY headers are *very long* (longer than payloads): 1024 symbols or more
- Sending short headers would require to transmit them with very high transmission power
- This is not permitted by the telecommunications regulator agencies (for example FCC in the USA)

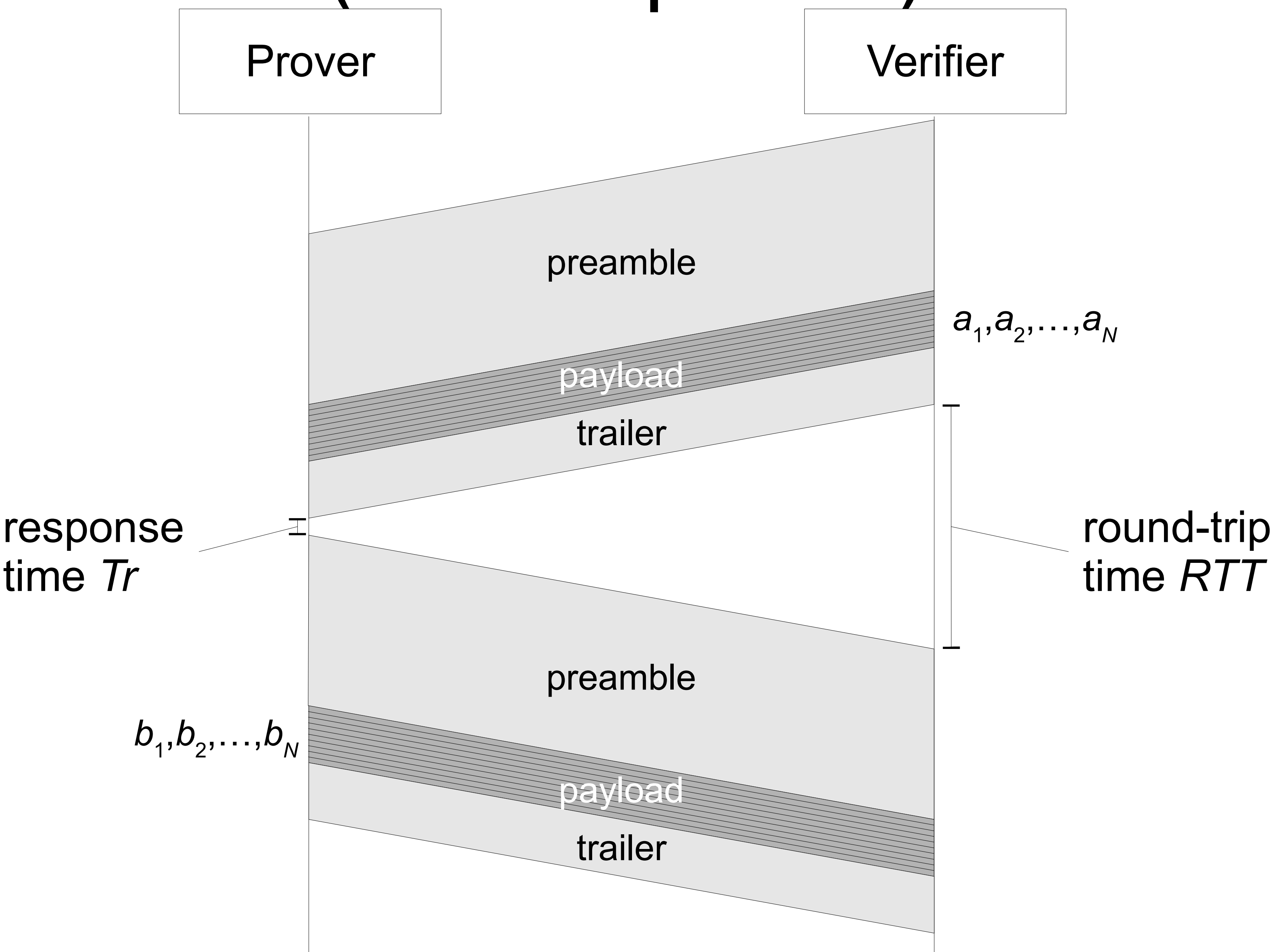
Frame-based distance bounding

- It is burdensome to send *single bits*
  - A very long PHY header for each bit!
- … and insecure:
  - A dishonest prover could leverage on the latency times to anticipate the transmission of the response bit

Frame-based distance bounding

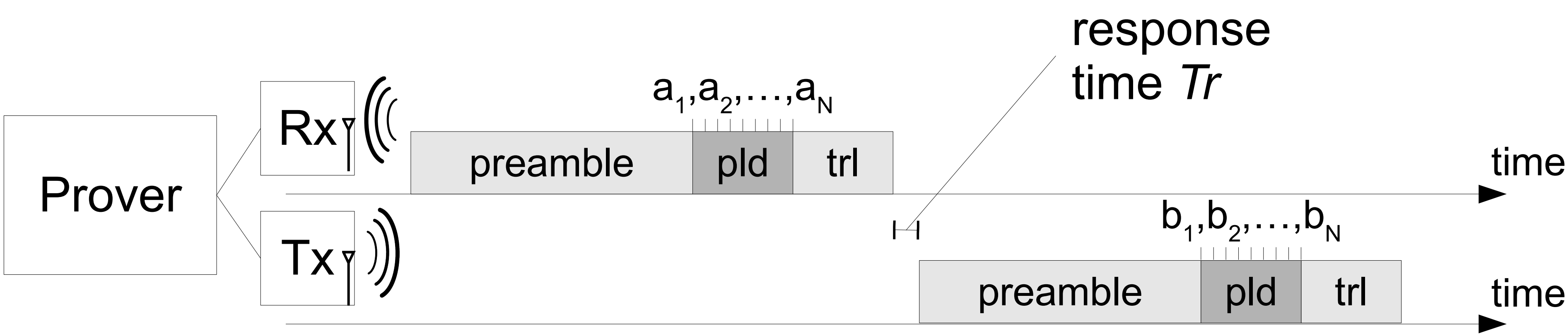
- The rapid bit exchange phase is replaced by a *frame exchange phase*
- Instead of performing *N* challenge-response rounds, we perform a single round with two *N*-bit frames

Frame exchange phase (honest prover)



Frame exchange phase (honest prover)

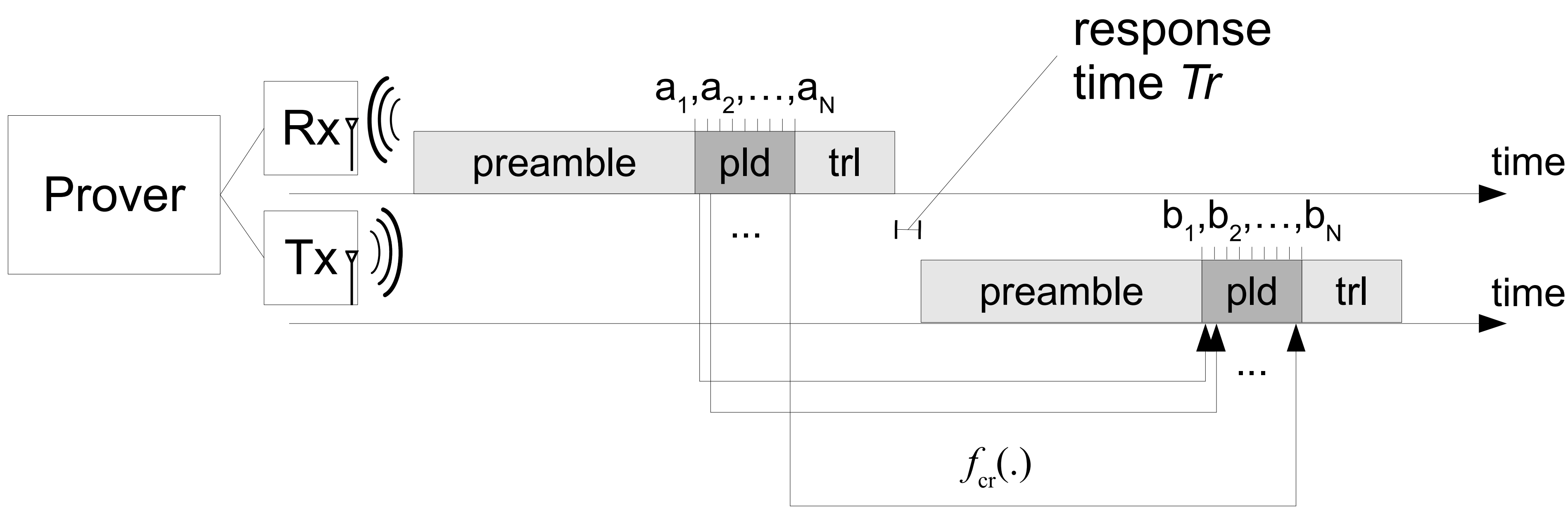
- *Timeline representation*:



- Can we use the same timing if we want to defend against *distance fraud* too?

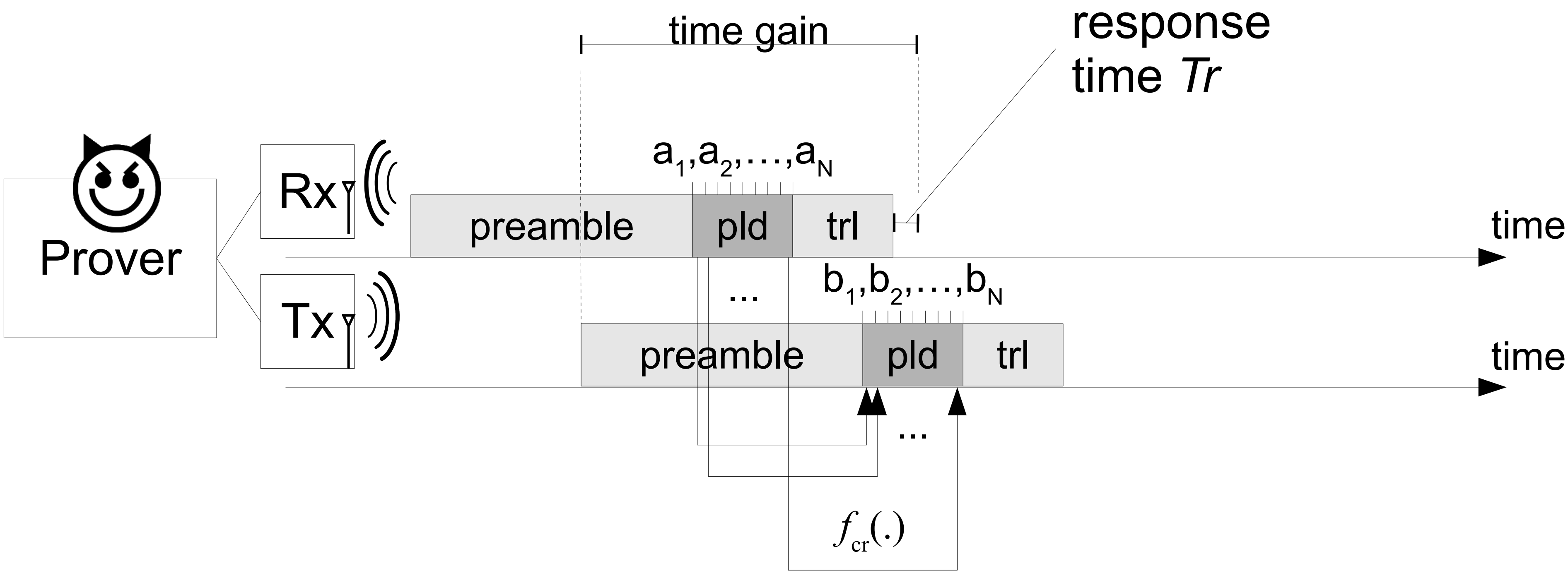
Frame exchange phase (honest prover)

- A frame-based challenge-response function is too complex to be computed on-the-fly



- A (classic) bit-based challenge-response function is used

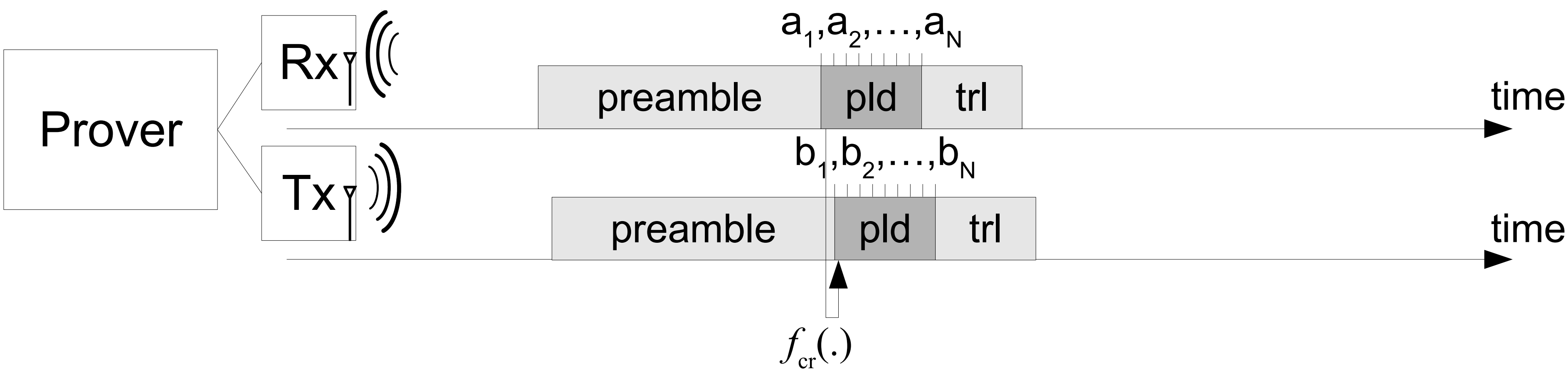
Distance fraud



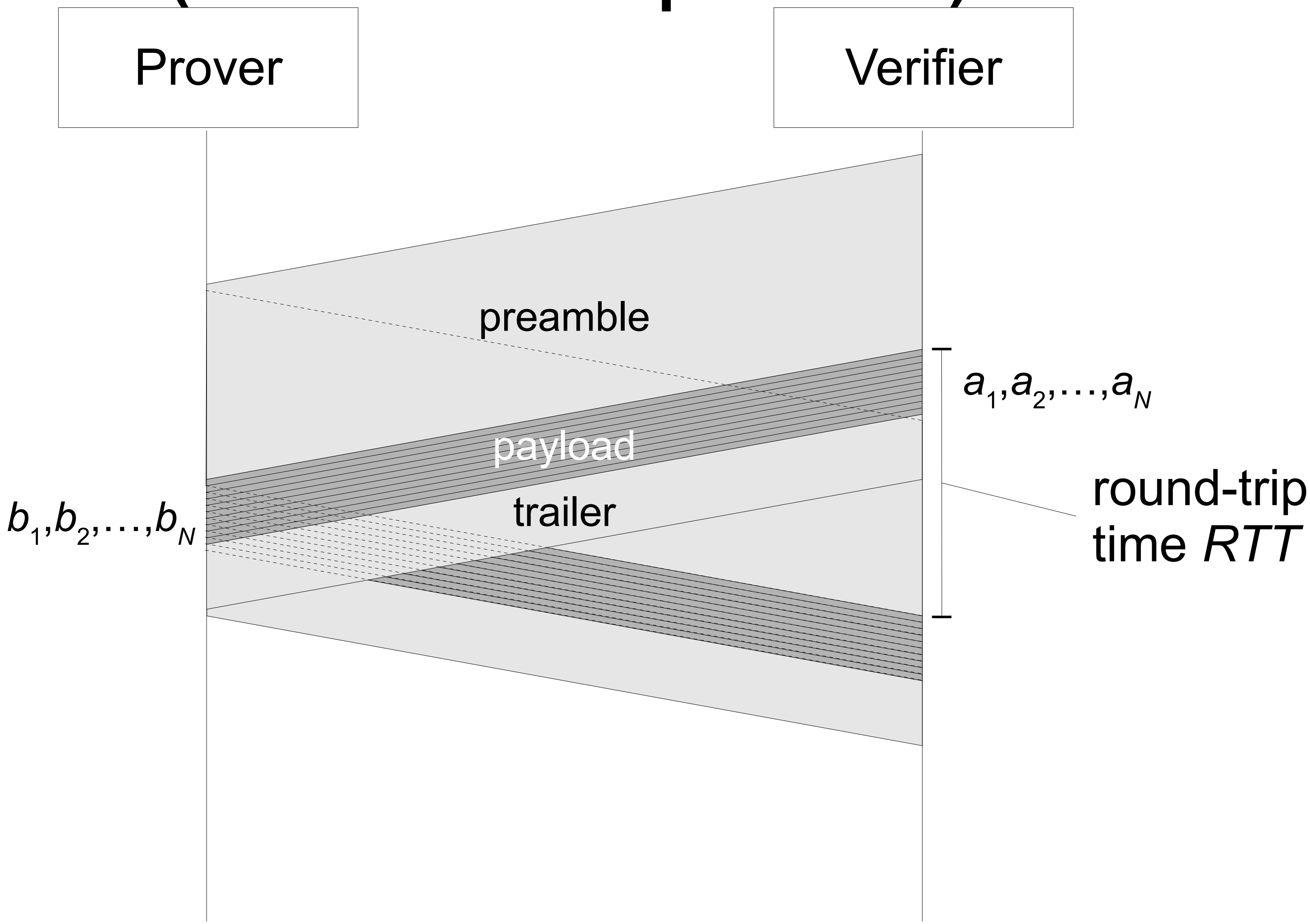
Distance fraud

- *Countermeasure*: the challenge-response phase is performed in a *full-duplex* way
- Each response bit is sent *just after* having received the correspondent challenge bit

Frame exchange phase (dishonest prover)



Frame exchange phase (dishonest prover)



References

- Stefan Brands and David Chaum. *"Distance-bounding protocols."* EUROCRYPT’93. Springer Berlin Heidelberg, 1994.
  - (Only sections 1 and 2)
- Gerhard Hancke and Markus Kuhn. *"An RFID distance bounding protocol."* SecureComm 2005. IEEE, 2005.

- Contact me for questions:

`pericle.perazzo@iet.unipi.it`