Figerprinting digital documents

survey

Gábor Tardos

Rényi Institute & Central European University

1. Government secrets

- Government meeting on Monday to discuss secret plans on hospital reorganizations in face of COVID-19

1. Government secrets

- Government meeting on Monday to discuss secret plans on hospital reorganizations in face of COVID-19
- All the details of the plan are front page news on Index on Tuesday

index

A bezárandó kórházi osztályok listája
- János kórház, belgyógyászat
- Margit kórház, szülészet
- …

2. Industry secrets

Director of engineering compony:

- Good news: We have just sold the thousandth copy of our video on how to build cratoons.

2. Industry secrets

Director of engineering compony:

- Good news: We have just sold the thousandth copy of our video on how to build cratoons.
- Bad news: this was the last one. Somebody uploaded it to YouTube – now anybody can watch it for free.

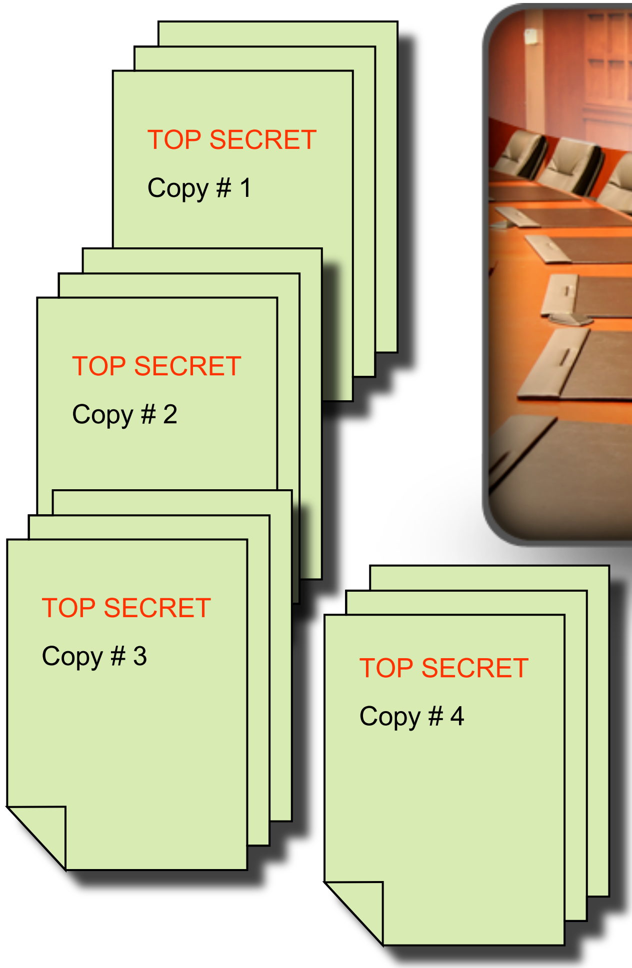
How to protect the secret

- Sue the medium (Index or YouTube) or at least make sure they stop sharing our information
- Sue the illegitimate end user (the guy who builds cratoons with our video but did not pay for it)

How to protect the secret

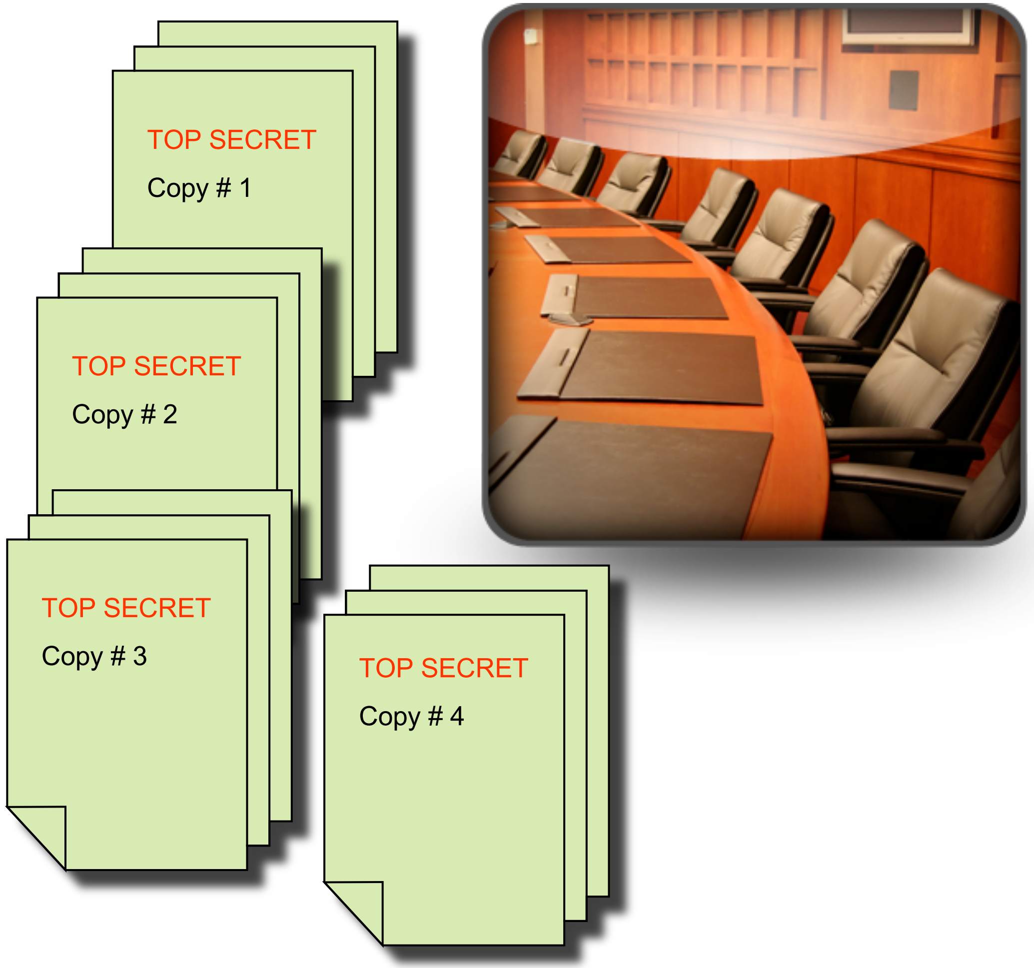
- Sue the medium (Index or YouTube) or at least make sure they stop sharing our information
- Sue the illegitimate end user (the guy who builds cratoons with our video but did not pay for it)
- In this talk: Find the legitimate user who illegally shared the secret (the cabinet member / one of the thousand customers who payed for the video)

Embed unique ID in every copy of document



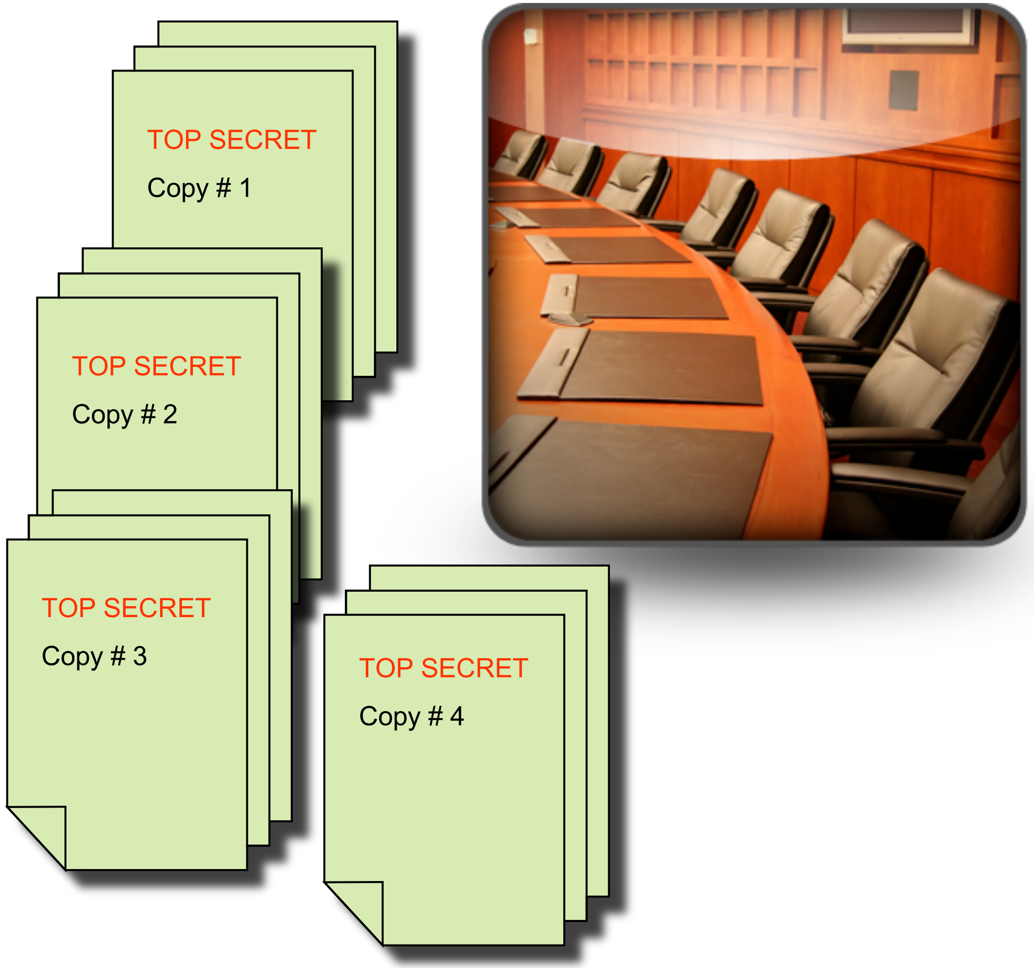
Embed unique ID in every copy of document

- Hide the embedded ID.
  If user finds it can remove the ID
  and make leaked copy untraceable.



Embed unique ID in every copy of document

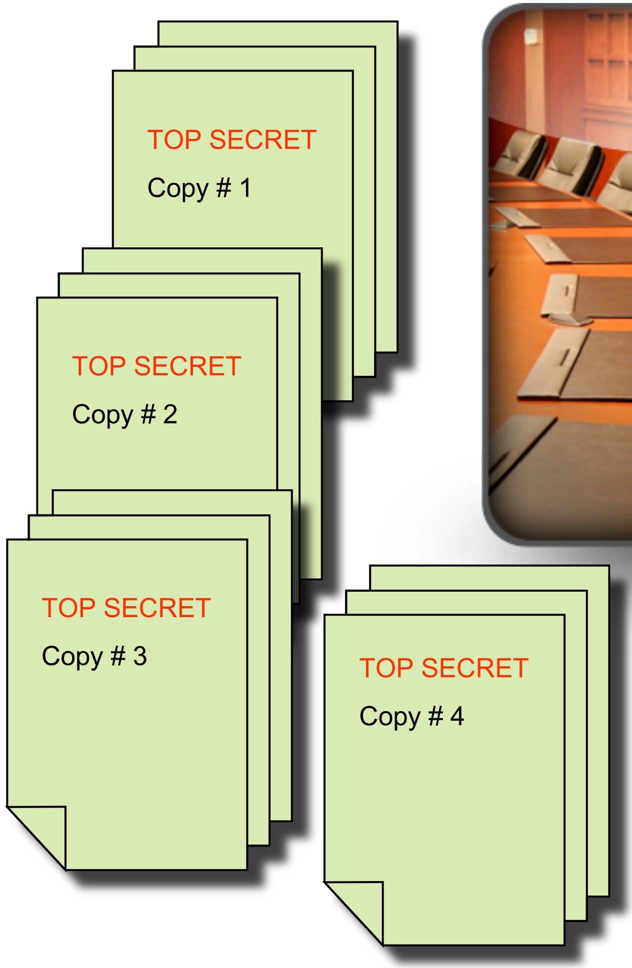
- Hide the embedded ID.
  If user finds it can remove the ID
  and make leaked copy untraceable.
- Easy for video / image / software
  (lots of irrelevant places to hide ID)
  harder (but doable) for text.
- Practical if number of legitimate users
  is small and they are known.



Embed unique ID in every copy of document

- Hide the embedded ID.
  If user finds it can remove the ID
  and make leaked copy untraceable.
- Easy for video / image / software
  (lots of irrelevant places to hide ID)
  harder (but doable) for text.
- Practical if number of legitimate users
  is small and they are known.

Example: Hollywood movies distributed to the members of the American Academy before the vote for the Oscars.



Example

Digital document:

0010010110101111101010110011010010001010001100110100111111

Example

Find irrelevant positions:

00100101101011111010 101 10011010010001 01000110011010011111

Example

Duplicate:

00100101101011111010 101 10011010010001 01000110011010011111
00100101101011111010 101 10011010010001 01000110011010011111
00100101101011111010 101 10011010010001 01000110011010011111
00100101101011111010 101 10011010010001 01000110011010011111
00100101101011111010 101 10011010010001 01000110011010011111
00100101101011111010 101 10011010010001 01000110011010011111
00100101101011111010 101 10011010010001 01000110011010011111

Example

Insert distinct code (ID) in every copy:

001001011010111110100101010011010010001001000110011010011111 1

0010010110101111101001010100110100100011010001100110100111111

0010010110101111101001011100110100100010010001100110100111111

0010010110101111101001011100110100100011010001100110100111111

0010010110101111101011010100110100100010010001100110100111111

0010010110101111101011010100110100100011010001100110100111111

0010010110101111101011011100110100100010010001100110100111111

Example

Insert distinct code (ID) in every copy:

00100101101011111010**0**101**0**10011010010001**0**010001100110100111111
00100101101011111010**0**101**0**10011010010001**1**010001100110100111111
00100101101011111010**0**101**1**10011010010001**0**010001100110100111111
00100101101011111010**0**101**1**10011010010001**1**010001100110100111111
00100101101011111010**1**101**0**10011010010001**0**010001100110100111111
00100101101011111010**1**101**0**10011010010001**1**010001100110100111111
00100101101011111010**1**101**1**10011010010001**0**010001100110100111111

- If code position remain hidden
- code is not changed
- leaking participant easily traced

No mathematics?!

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{b-a}{2}\xi + \frac{b+a}{2}\right)\left(\frac{b-a}{2}d\xi\right)$$

$$= \frac{b-a}{2}\int_{-1}^{1} g(\xi)d\xi = \frac{b-a}{2}\sum_{k=1}^{n} w(\xi_k)g(\xi_k) + R_n(\xi)$$

$$= \frac{b-a}{2}\sum_{k=1}^{n} w(\xi_k)f\left(\frac{b-a}{2}\xi_k + \frac{b+a}{2}\right) + R_n(\xi)$$

where $\xi = \frac{2x-b-a}{b-a}$, i.e., $x = \frac{b-a}{2}\xi + \frac{b+a}{2}$, $-1 < \xi < 1$,

ξ_k is the kth zero of $P_n(\xi)$,

$$w(\xi_k) = \frac{2}{\left(1-\xi_k^{\,2}\right)\left[P_n{}'(\xi_k)\right]^2},$$

$$g(\xi) = f\left(\frac{b-a}{2}\xi_k + \frac{b+a}{2}\right),$$

$$R_n(\xi) = \frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3}g^{(2n)}(\xi).$$

No mathematics?!

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{b-a}{2}\xi + \frac{b+a}{2}\right)\left(\frac{b-a}{2}d\xi\right)$$

$$= \frac{b-a}{2}\int_{-1}^{1} g(\xi)d\xi = \frac{b-a}{2}\sum_{k=1}^{n} w(\xi_k)g(\xi_k) + R_n(\xi)$$

$$= \frac{b-a}{2}\sum_{k=1}^{n} w(\xi_k) f\left(\frac{b-a}{2}\xi_k + \frac{b+a}{2}\right) + R_n(\xi)$$

where $\xi = \frac{2x-b-a}{b-a}$, i.e., $x = \frac{b-a}{2}\xi + \frac{b+a}{2}$, $-1 < \xi < 1$,

ξ_k is the kth zero of $P_n(\xi)$,

$$w(\xi_k) = \frac{2}{\left(1-\xi_k{}^2\right)\left[P_n{}'(\xi_k)\right]^2},$$

$$g(\xi) = f\left(\frac{b-a}{2}\xi_k + \frac{b+a}{2}\right),$$

$$R_n(\xi) = \frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} g^{(2n)}(\xi).$$

it’s coming…

Collusion attack

Two (or more) participant compare copies:

```
  00100101101011111010010100100110100100010010001100110100111111
  00100101101011111010010100100110100100011010001100110100111111
→ 00100101101011111010010110100110100100010010001100110100111111
  00100101101011111010010110100110100100011010001100110100111111
→ 00100101101011111010110100100110100100010010001100110100111111
  00100101101011111010110100100110100100011010001100110100111111
  00100101101011111010110110100110100100010010001100110100111111
```

Collusion attack

Two (or more) participant compare copies:

→ 0010010110101111101000101110011010010001001000110011010011111

→ 00100101101011111010110101001101001000100100011001101001111111

Differences between documents:

Collusion attack

Two (or more) participant compare copies:

0010010110101111101001011100110100100010010001100110100111111

0010010110101111101011010100110100100010010001100110100111111

Differences between documents:

These positions of the code can be altered arbitrarily:

makes tracing much harder (and more interesting!)

Collusion attack

Two (or more) participant compare copies:

→ 00100101101011111010010111001101001000100100011001101001111111

→ 00100101101011111010110101001101001000100100011001101001111111

Some positions of code may remain hidden

Differences between documents:

These positions of the code can be altered arbitrarily:

makes tracing much harder (and more interesting!)

Collusion attack

Two (or more) participant compare copies:

→ 0010010110101111101001011100110100100010010001100110100111111

→ 0010010110101111101011010100110100100010010001100110100111111

Differences between documents:

These positions of the code can be altered arbitrarily:

makes tracing much harder (and more interesting!)

Some positions of code may remain hidden

tracing must be based on these

Boneh-Shaw fingerprinting model

Limited number of malicious participants (the pirates) collaborate to forge untraceable copy of document.

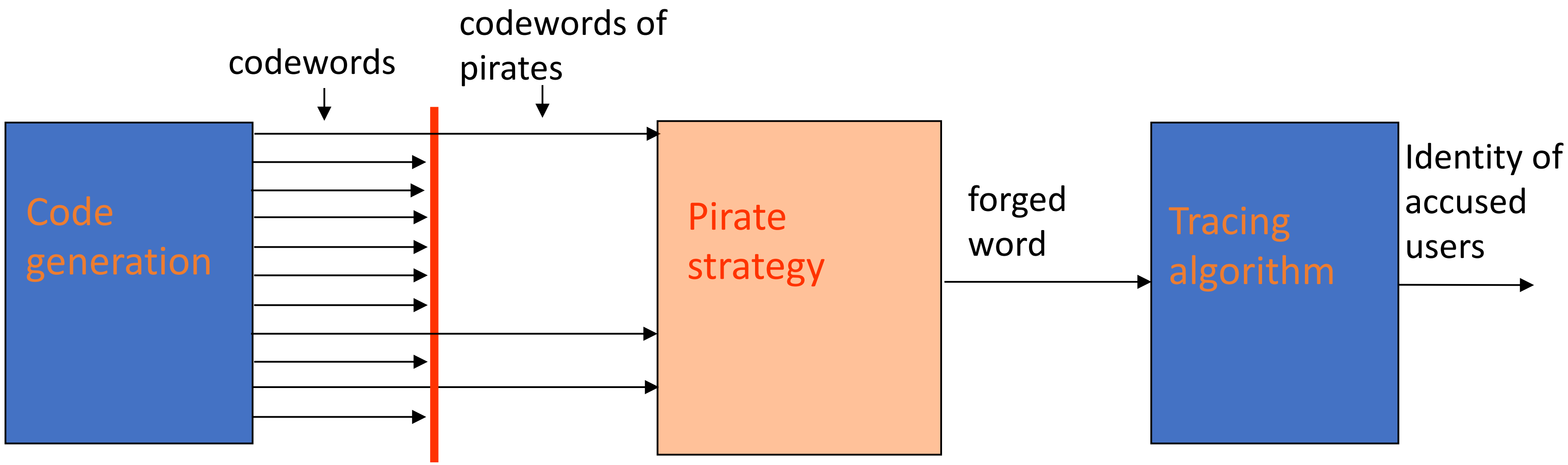
Boneh-Shaw fingerprinting model

Limited number of malicious participants (the pirates) collaborate to forge untraceable copy of document.

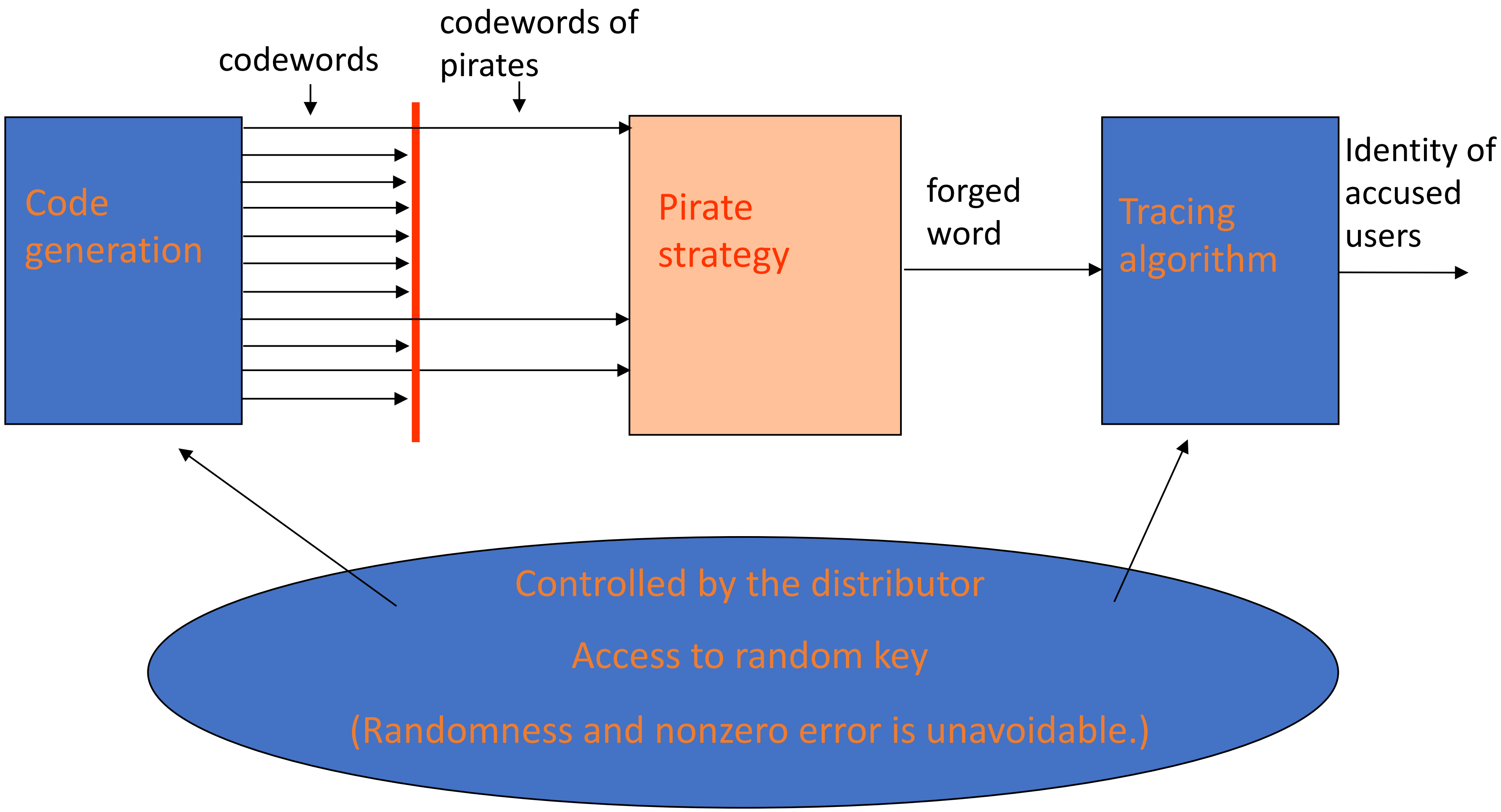
They don’t find / cannot change positions of code that agrees in each codeword they have: the Marking Assumption.

They are not restricted in their output in any other way.

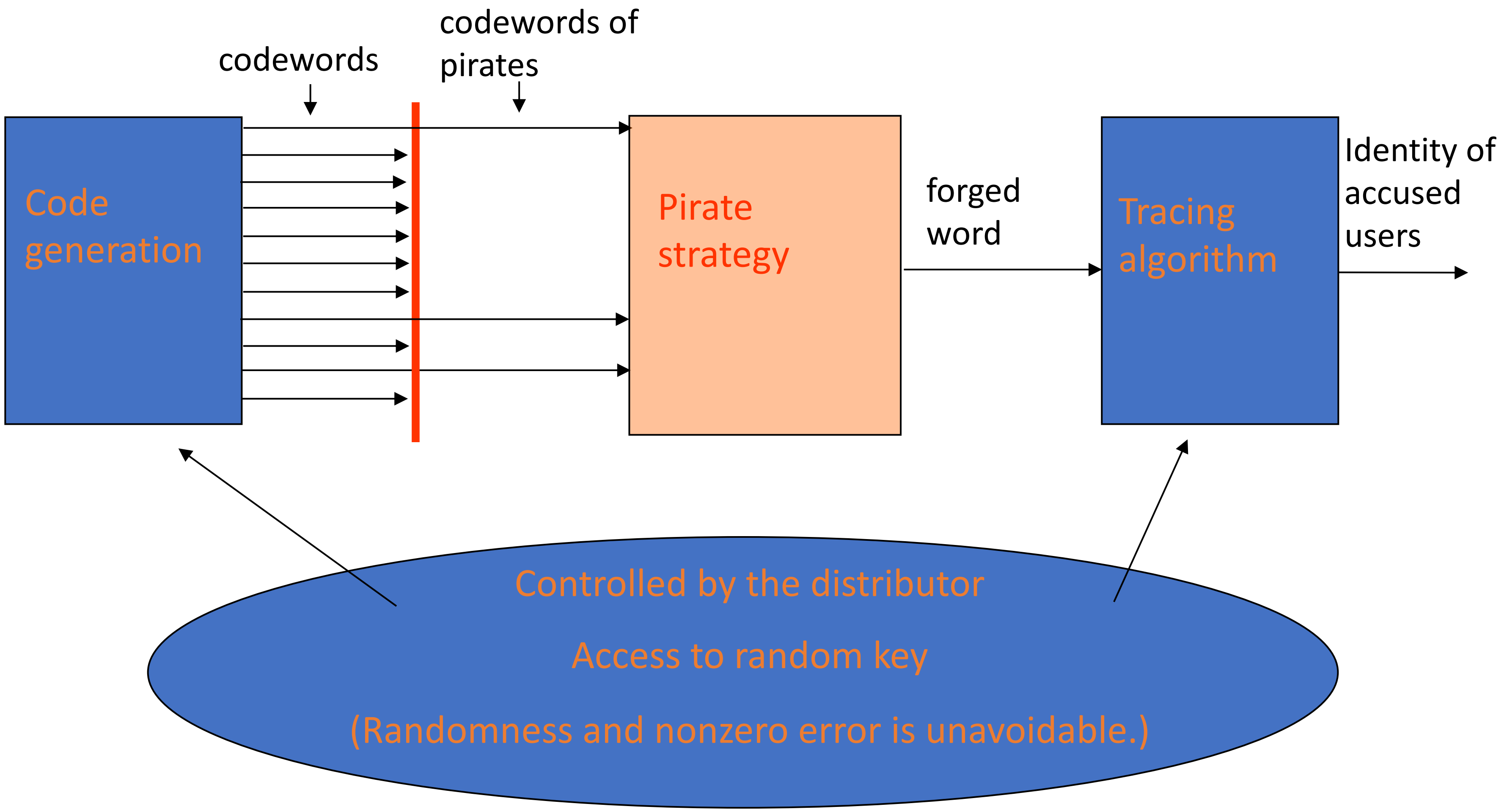
Boneh-Shaw fingerprinting model



Boneh-Shaw fingerprinting model



Boneh-Shaw fingerprinting model

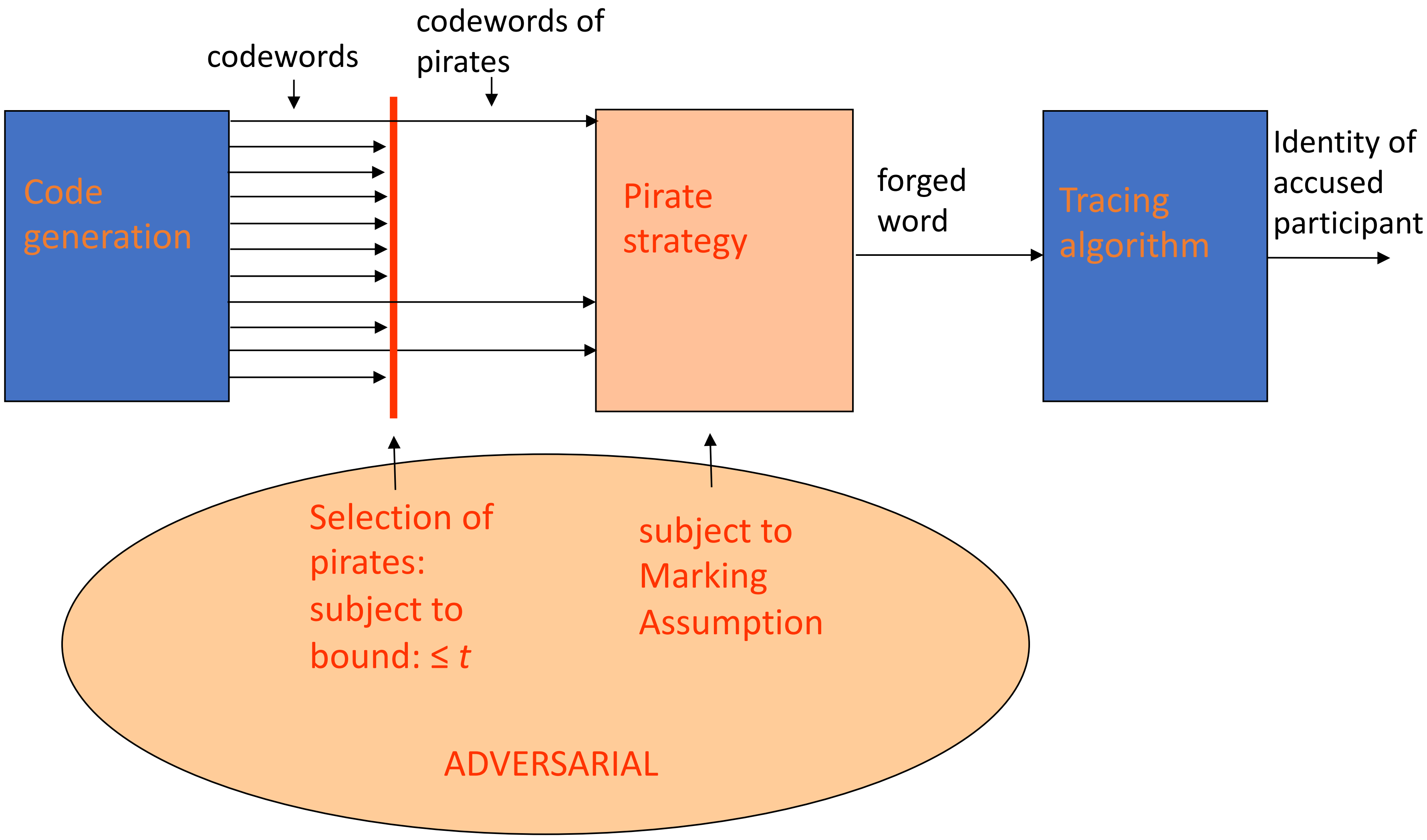


Goal of the distributor: accuse pirate(s)

Error: an innocent user accused

Fail: no pirate is accused

Boneh-Shaw fingerprinting model



Parameters of fingerprinting code:

- number of participants: *N* considered large
- max number of pirates: *t* considered a constant
- length of code: *n*
- size of alphabet: *s* s=2 for binary, s>2 for non-binary
- worst case error / fail probability

Parameters of fingerprinting code:

- number of participants: N considered large
- max number of pirates: t considered a constant
- length of code: n
- size of alphabet: s s=2 for binary, s>2 for non-binary
- worst case error / fail probability
- rate: $R = \log N / n$, $N = 2^{Rn}$ $R = \log s$ for no collision ($t = 1$), $R < \log s$ otherwise

Parameters of fingerprinting code:

- number of participants: N considered large
- max number of pirates: t considered a constant
- length of code: n
- size of alphabet: s s=2 for binary, s>2 for non-binary
- worst case error / fail probability
- rate: $R = \log N / n$, $N = 2^{Rn}$ $R = \log s$ for no collision ($t = 1$), $R < \log s$ otherwise

Simplification:

Maximize rate subject to error probability going to zero as length grows.

Maximal rate = t-fingerprinting capacity (also depends on s)

Constructions, bounds

Boneh-Shaw 1988: t-secure binary fingerprinting codes with rate: $R = \Omega(t^{-4})$

bound on t-fingerprinting capacity: $O(t^{-1})$

T. 2003, 2008: bias code generation, linear accusation: $R = t^{-2} / 100$

bound on t-fingerprinting capacity : $O(t^{-2})$

Constructions, bounds

Boneh-Shaw 1988: t-secure binary fingerprinting codes with rate: $R = \Omega(t^{-4})$

bound on t-fingerprinting capacity: $O(t^{-1})$

T. 2003, 2008: bias code generation, linear accusation: $R = t^{-2} / 100$

bound on t-fingerprinting capacity : $O(t^{-2})$

construction is binary, but bound applies for arbitrary alphabet size:

no need to ever to consider non-binary alphabets or more complicated codes???

Constructions, bounds

Boneh-Shaw 1988: t-secure binary fingerprinting codes with rate: $R = \Omega(t^{-4})$

bound on t-fingerprinting capacity: $O(t^{-1})$

T. 2003, 2008: bias code generation, linear accusation: $R = t^{-2} / 100$

bound on t-fingerprinting capacity : $O(t^{-2})$

construction is binary, but bound applies for arbitrary alphabet size:

no need to ever to consider non-binary alphabets or more complicated codes???

Huge constant factor between lower and upper bound became subject of intense research:

Skoric-Katzenbeisser-Celik, Skoric-Vladimirova-Celik-Talastra, Blayer-Tassa

While others focused on the capacity for small constant values of t:

Anthapadmanabhan-Barg, Anthapadmanabhan-Barg-Dumer, Barg-Blakeley

Newer constructions, bounds

Amiri-T.: t-secure binary fingerprinting codes with much improved rates:

conjectured to achieve t-fingerprinting capacity for any t.

Improved bound on binary t-fingerprinting capacity.

Both rate of construction and bound is $(1/(2\ln 2) + o(1))\, t^{-2}$

Asymptotical agreement, but do not agree for any fixed t.

Huang-Moulin, Moulin: Similar construction for a much broader class of fingerprinting problems

simpler fingerprinting (T.)

Bias code generation

- find biases $0 < bi < 1, \ i = 1, 2, \dots, n,$ i.i.d from fix distribution D;
- choose bit *i* of binary codeword *x* with bias b_i: $\Pr[x_i = 1] = b_i$;
- every bit of every codeword independent (given the biases)

simpler fingerprinting (T.)

Bias code generation

- find biases $0 < bi < 1, \ i = 1, 2, \ldots, n,$ i.i.d from fix distribution D;
- choose bit *i* of binary codeword *x* with bias b_i: $\Pr[x_i = 1] = b_i$;
- every bit of every codeword independent (given the biases)

linear tracing

- given forged word y accuse user with codeword x if

$$\sum_{i=1}^{n} f(x_i, y_i, b_i) > T$$

simpler fingerprinting (T.)

Bias code generation

- find biases $0 < bi < 1, \; i = 1, 2, \ldots, n$, i.i.d from fix distribution D;
- choose bit *i* of binary codeword *x* with bias b_i: $\Pr[x_i = 1] = b_i$;
- every bit of every codeword independent (given the biases)

linear tracing

- given forged word y accuse user with codeword x if

$$\sum_{i=1}^{n} f(x_i, y_i, b_i) > T$$

Optimize
– distribution D
– function f
– threshold T

improved fingerprinting (Amiri-T.)

- Bias code generation
- More complex tracing

improved fingerprinting (Amiri-T.)

- Bias code generation
- More complex tracing:
  Consider each subset of $\leq t$ users as potential set of pirates,
  accuse the smallest set that could reasonably produce the pirated output

improved fingerprinting (Amiri-T.)

- Bias code generation
- More complex tracing:
  Consider each subset of $\leq t$ users as potential set of pirates,
  accuse the smallest set that could reasonably produce the pirated output

based on
mutual information
between codewords
and the forged word

improved fingerprinting (Amiri-T.)

- Bias code generation
- More complex tracing:
  Consider each subset of $\leq t$ users as potential set of pirates,
  accuse the smallest set that could reasonably produce the pirated output

based on
mutual information
between codewords
and the forged word

Optimization via
equilibrium in 2-person
information theoretic game

improved fingerprinting (Amiri-T.)

- Bias code generation
- More complex tracing:
  Consider each subset of $\leq t$ users as potential set of pirates,
  accuse the smallest set that could reasonably produce the pirated output

based on
mutual information
between codewords
and the forged word

Optimization via
equilibrium in 2-person
information theoretic game

Advantage:
near-optimal
rate
Disadvantage:
very slow
tracing

GOAL

Combine:

- near-optimal rate
- efficient (linear time) tracing

GOAL

Combine:

- near-optimal rate
- efficient (linear time) tracing

First step: doable for $t = 2$ pirates

?????? for $t > 2$???????

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

Diana picks distribution D on S

Pierre picks conditional distribution $C = (y \mid x_1,\ldots,x_t)$

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

Diana picks distribution D on S

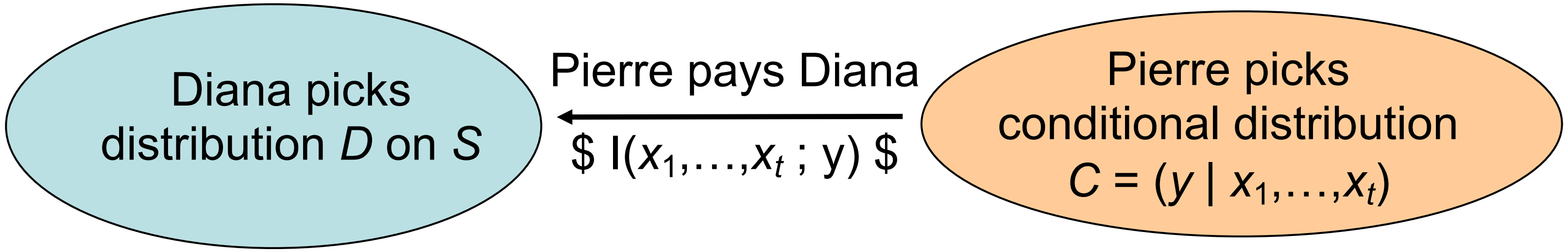
Pierre picks conditional distribution $C = (y \mid x_1,\ldots,x_t)$

A probability space is created with $x_1,\ldots,x_t$
i.i.d. letters from S according to D
and y is another letter from S
generated according to C.

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

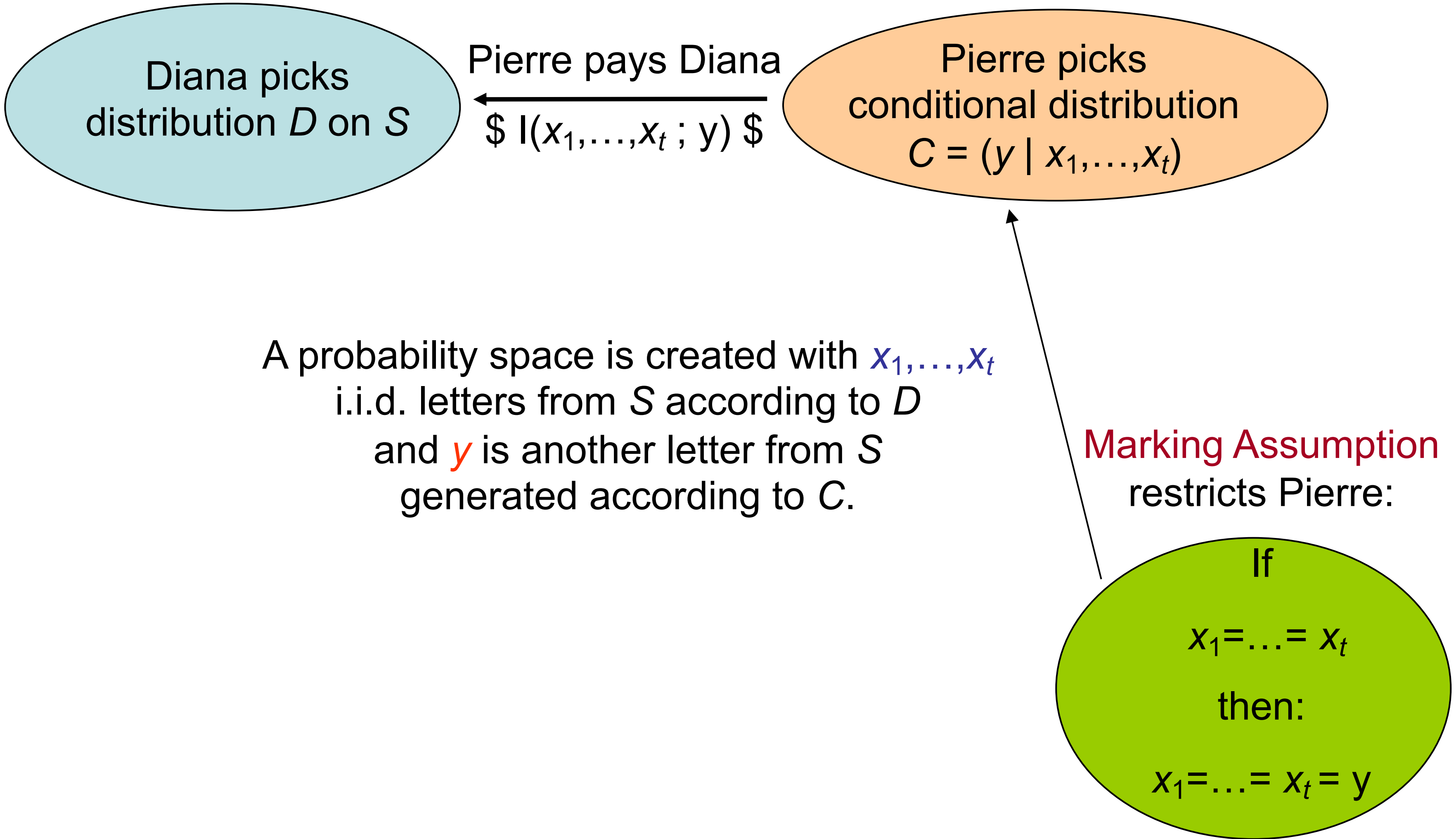


A probability space is created with $x_1,\ldots,x_t$
i.i.d. letters from S according to D
and y is another letter from S
generated according to C.

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.



The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

Diana picks distribution D on S

Pierre pays Diana

$ $I(x_1,\ldots,x_t\,;\, y)$ $

Pierre picks conditional distribution $C = (y \mid x_1,\ldots,x_t)$

A probability space is created with $x_1,\ldots,x_t$ i.i.d. letters from S according to D and y is another letter from S generated according to C.

Marking Assumption restricts Pierre:

If

$x_1 = \ldots = x_t$

then:

$x_1 = \ldots = x_t = y$

Moulin considers other restrictions in place of the Marking Assumption:

Different versions of fingerprinting

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

Diana picks distribution D on S

Pierre pays Diana

$ I($x_1,\ldots,x_t$; y) $

Pierre picks conditional distribution $C = (y \mid x_1,\ldots,x_t)$

The Minimax Theorem states the existence of saddle point equilibrium for mixed strategies.

Does not hold for all infinite games.

Marking Assumption restricts Pierre:

If

$x_1 = \ldots = x_t$

then:

$x_1 = \ldots = x_t = y$

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

Diana picks distribution D on S

Pierre pays Diana

$ $\mathrm{I}(x_1,\ldots,x_t\,;\,y)$ $

Pierre picks conditional distribution $C = (y \mid x_1,\ldots,x_t)$

Marking Assumption restricts Pierre:

If

$x_1 = \ldots = x_t$

then:

$x_1 = \ldots = x_t = y$

The Minimax Theorem states the existence of saddle point equilibrium for mixed strategies.

Does not hold for all infinite games, but this is a convex game:

- Minimax holds

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

Diana picks distribution D on S

Pierre pays Diana

$ I$(x_1,\ldots,x_t ; y)$ $

Pierre picks conditional distribution $C = (y \mid x_1,\ldots,x_t)$

The Minimax Theorem states the existence of saddle point equilibrium for mixed strategies.

Does not hold for all infinite games, but this is a convex game:

- Minimax holds
- Pierre's optimal strategy is deterministic
- Diana's optimal strategy is a randomized, but over just a few possible D.

Marking Assumption restricts Pierre:

If

$x_1 = \ldots = x_t$

then:

$x_1 = \ldots = x_t = y$

The continuous game

Players: Diana and Pierre.

Parameters: number $t \geq 2$ and finite alphabet S.

Diana picks distribution D on S

Pierre pays Diana

$ I($x_1,\ldots,x_t$; y) $

Pierre picks conditional distribution $C = (y \mid x_1,\ldots,x_t)$

The Minimax Theorem states the existence of saddle point equilibrium for mixed strategies.

Does not hold for all infinite games, but this is a convex game:

- Minimax holds
- Pierre's optimal strategy is deterministic
- Diana's optimal strategy is a randomized, but over just a few possible D.
- Rate achieved in fingerprinting = (value of this game)/t

Marking Assumption restricts Pierre:

If

$x_1 = \ldots = x_t$

then:

$x_1 = \ldots = x_t = y$