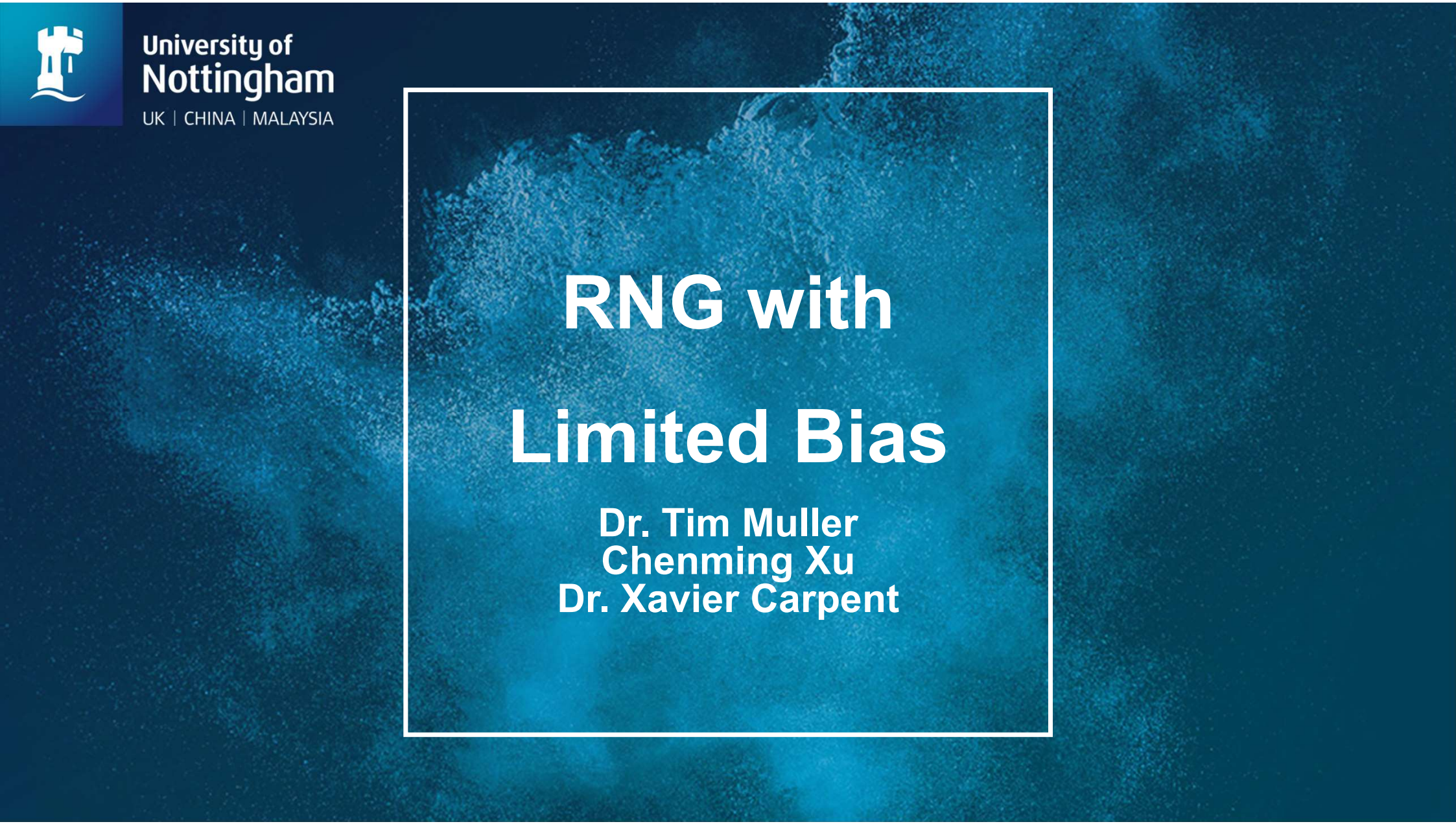

University of
Nottingham
UK | CHINA | MALAYSIA
RNG with
Limited Bias
Dr. Tim Muller
Chenming Xu
Dr. Xavier Carpent

University of Nottingham
UK | CHINA | MALAYSIA

About Me

- Background in formal methods and the application of computational trust to security.

About Me

- Background in formal methods and the application of computational trust to security.
  - I like proving things

About Me

- Background in formal methods and the application of computational trust to security.
  - I like proving things
  - I like measuring things

About Me

- Background in formal methods and the application of computational trust to security.
  - I like proving things
  - I like measuring things
  - I hate when people slap numbers on things and call it a day.

About Me

- Background in formal methods and the application of computational trust to security.
  - I like proving things
  - I like measuring things
  - I hate when people slap numbers on things and call it a day.

About Sjouke

- Sjouke was my PhD supervisor in Luxembourg from 2009-2013.
  - What did I learn during my second month?

About Sjouke

- Sjouke was my PhD supervisor in Luxembourg from 2009-2013.
  - What did I learn during my second month?

Artemis II mission, 2026
by Victor Glover
“Far side of the moon”

About Sjouke

- Sjouke was my PhD supervisor in Luxembourg from 2009-2013.
  - What did I learn during my second month?

Artemis II mission, 2026
by Victor Glover
“Far side of the moon”

Christmas party, Uni Lu, 2009
by Patrick Schweitzer

About Sjouke

- Sjouke was my PhD supervisor in Luxembourg from 2009-2013.
  - What did I learn during my second month?

Artemis II mission, 2026
by Victor Glover
“Far side of the moon”

Christmas party, Uni Lu, 2009
by Patrick Schweitzer

“Far side of the moon”

University of Nottingham
UK | CHINA | MALAYSIA

About Sjouke

- Sjouke was my PhD supervisor in Luxembourg from 2009-2013.
  - What did I learn during my second month?

Artemis II mission, 2026
by Victor Glover
“Far side of the moon”

Christmas party, Uni Lu, 2009
by Patrick Schweitzer

“Far side of the moon”

About Sjouke

- Research should be **Interesting**

About Sjouke

- Research should be **Interesting**
- Research should be **RIGOROUS**

About Sjouke

- Research should be **Interesting**
- Research should be **RIGOROUS**
- Research should be **grounded**

About Sjouke

- Research should be **Interesting**
- Research should be **RIGOROUS**
- Research should be **grounded**
- Research should be **FUN**


University of
Nottingham
UK | CHINA | MALAYSIA
Let’s Play Cards!

Game 1: The unfair Coinflip

Step 1:

Sjouke picks a colour card: Black or Red.

Step 2:

Trusted party Reveals Card

Game 1: The unfair Coinflip

Step 1:

Sjouke picks a colour card: Black or Red.

Step 2:

Trusted party Reveals Card

Step 3:

Tim always wins once the other colour is revealed

Game 1: The unfair Coinflip

Step 1:

Sjouke picks a colour card: Black or Red.

Step 2:

Trusted party Reveals Card

Step 3:

Tim always wins once the other colour is revealed

Unfair: Trusted party actively cheated.
Solution: Fix the deck (e.g. *VRF*)

Game 2: The unfair Race

Step 1:

Sjouke picks a specific card.

Step 2:

Tim picks a specific card.

Step 3:

Trusted party reveals cards from top down, until Tim's or Sjouke's card comes up.

Game 2: The unfair Race

Step 1:

Sjouke picks a specific card.

Step 2:

Tim picks a specific card.

Step 3:

Trusted party reveals cards from top down, until Tim's or Sjouke's card comes up.

Step 4:

Tim always wins using insider knowledge

Game 2: The unfair Race

Step 1:

Sjouke picks a specific card.

Step 2:

Tim picks a specific card.

Step 3:

Trusted party reveals cards from top down, until Tim's or Sjouke's card comes up.

Step 4:

Tim always wins using insider knowledge

Unfair: Trusted party passively cheated
Solution: No TTP, only MPC

Game 3: The unfair Die throw

Step 1:

Sjouke picks 5 dice values.

Step 2:

Tim gets remaining value

Step 3:

Throw a die

Game 3: The unfair Die throw

Step 1:

Sjouke picks 5 dice values.

Step 2:

Tim gets remaining value

Step 3:

Throw a die

... Does anyone have a die?

No? OK, we'll do MPC

Game 3a: The unfair Emulated Die Throw

Step 1&2: Sjouke picks 5 values, Tim gets 1 winning value.

Step 3:

Sjouke selects a card from [1..6], call it s

Step 4:

Tim selects a card from [1..6], call it t

Step 5:

Emulated throw is $(s + t) \bmod 6$

Game 3a: The unfair Emulated Die Throw

Step 1&2: Sjouke picks 5 values, Tim gets 1 winning value.

Step 3:

Sjouke selects a card from [1..6], call it s

Step 4:

Tim selects a card from [1..6], call it t

Step 5:

Emulated throw is $(s + t) \bmod 6$

Step 6:

Tim always wins (due to knowledge asymmetry)

Game 3b: The unfair Committed Emulated Die Throw

Step 1&2: Sjouke picks 5 values, Tim gets 1 winning value.

Step 3&4: Cards are selected *face down*.

Step 5:

Sjouke reveals card

Game 3b: The unfair Committed Emulated Die Throw

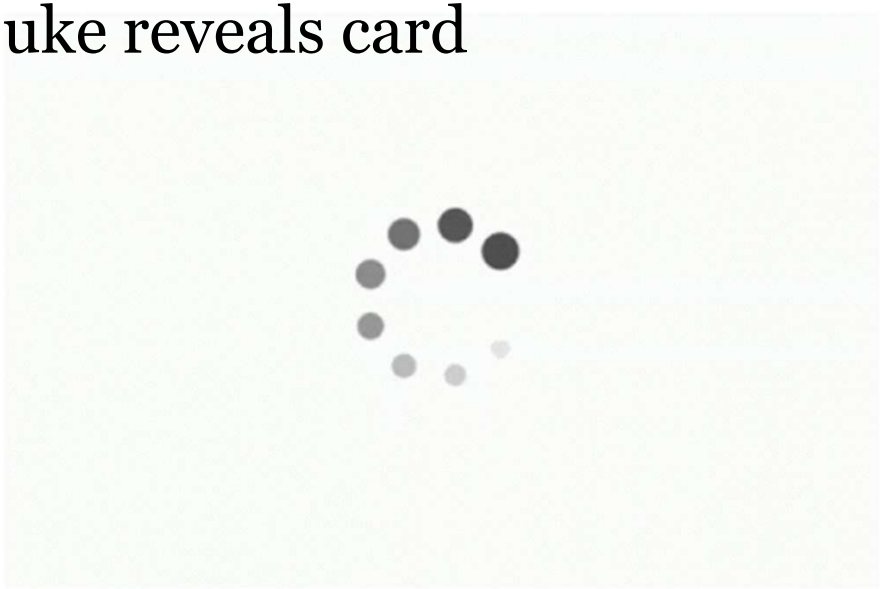
Step 1&2: Sjouke picks 5 values, Tim gets 1 winning value.

Step 3&4: Cards are selected *face down*.

Step 5:

Sjouke reveals card

Step 6:



Game 3b: The unfair Committed Emulated Die Throw

Step 1&2: Sjouke picks 5 values, Tim gets 1 winning value.

Step 3&4: Cards are selected *face down*.

Step 5:

Sjouke reveals card

Step 6:

Tim reveals card

Step 7:

Emulated throw is $(s + t) \bmod 6$

Step 8:

Tim always wins (i.e. wins every completed game)

Game 3c: The 4-way unfair Committed Emulated Die Throw with Elimination

1&2: Sjouke picks 5 values, rest gets the other value

3&4: All players pick card face down.

5&6: Players reveal card or get eliminated

Step 7:

Final value is computed from the revealed values.

Game 3c: The 4-way unfair Committed Emulated Die Throw with Elimination

1&2: Sjouke picks 5 values, rest gets the other value

3&4: All players pick card face down.

5&6: Players reveal card or get eliminated

Step 7:

Final value is computed from the revealed values.

Step 8:

Tim's coalition (always) wins

Game 4: The 4-way unfair Cmmt Emlt 13-sided Die Throw w/ Elimination

1&2: Sjouke picks 12 values, rest gets the other value

3&4: All players pick card face down.

5&6: Players reveal card or get eliminated

Step 7:

Final value is computed from the revealed values.

Game 4: The 4-way unfair Cmmt Emlt 13-sided Die Throw w/ Elimination

1&2: Sjouke picks 12 values, rest gets the other value

3&4: All players pick card face down.

5&6: Players reveal card or get eliminated

Step 7:

Final value is computed from the revealed values.

Step 8:

Tim's coalition wins with probability $^{8}/_{13}$

Game 4: The 4-way unfair Cmmt Emlt 13-sided Die Throw w/ Elimination

1&2: Sjouke picks 12 values, rest gets the other value

3&4: All players pick card face down.
5&6: Players reveal card or get eliminated

Step 7:

Final value is computed from the revealed values.

Step 8:

Tim's coalition wins with probability $^{8}/_{13}$

+ Finite (8-fold) advantage to cheater
– Exponential (2^3) advantage to cheater

Goal for Somewhat Biased BFT-RNG

- Don’t want randomness beacons/TTPS
- Don’t want reliance on majorities/stakes
- Don’t want expensive cryptography (no timelock puzzles
- But then: Fair Byzantine-Fault Tolerant RNG is impossible..

University of Nottingham
UK | CHINA | MALAYSIA

Goal for Somewhat Biased BFT-RNG

- Don’t want randomness beacons/TTPS
- Don’t want reliance on majorities/stakes
- Don’t want expensive cryptography (no timelock puzzles
- But then: Fair Byzantine-Fault Tolerant RNG is impossible..

Goal for Somewhat Biased BFT-RNG

- Don’t want randomness beacons/TTPS
- Don’t want reliance on majorities/stakes
- Don’t want expensive cryptography (no timelock puzzles
- But then: Fair Byzantine-Fault Tolerant RNG is impossible..

- Instead, allow the cheater a *limited* advantage.
  - What is the asymptotic advantage?
  - What is the precise advantage?


University of
Nottingham
UK | CHINA | MALAYSIA
Matching Problem
Motivating Problem

Motivation for Matching Problem

- Toy problem:
  - Users should be able to change *pseudonymous* public key at will, but not be able to use multiple pseudonyms/keys at a time.
- Idea; *mixing*:
  - Allow any group of k users to publish (e.g. on a public ledger) a list of their old public keys, a list of new public keys, signed by their old keys.
- Problems:
  - If all other users are (colluding) attackers, then the mix was pointless.
  - If some users are attackers, then they can block the mix.
- We need to quantify the probabilities.

Matching Protocol

- There are n users and *you*.
  - Of these n users, m are attackers.
- A *Matching Protocol* puts you in a group of k users.
  - If all $k - 1$ other users are honest, then you win.
    - And gain k anonymity.
  - If not all other users are honest, then you don't win.
    - Either an attacker can block, or deduce linkage.
    - No gains, but only loss is waste of time/resource.

- *Fair RNG* protocol winning prob: $\frac{(n-m)!\cdot(n-k+1)!}{(n-m-k+1)!\cdot n!} \approx \left(\frac{n-m}{n}\right)^{k-1}$

Paranoid

- The games made us paranoid…
  - Set m as large as possible.
  - If $m = n$, then we are doomed no matter what.
  - What if $m = n - 1$?
- With this choice of m, previous formula simplifies:
  - $\frac{1}{n}$ if $k = 2$
  - 0 if $k > 2$
  - So only pairing protocols ($k = 2$) work in paranoid setting.
- Let's investigate some Protocols!

Don't Panic


University of
Nottingham
UK | CHINA | MALAYSIA
Buckets
It’s pronounced bouquets

Users Commit to Buckets

- Separation of concerns: what another user's random value is should not affect my pairing.

1. Each user commits to a *bucket* (at random).
2. Each user reveals their bucket.
3. Your match is the user(s) in your bucket.

- Win iff there is an honest user in your bucket and there is no attacker in your bucket.

Quantitative Analysis of Buckets

- If there are k buckets, n users and $m = n - 1$ attackers:
  - Probability of ending up with honest user: $^{1}/_{k}$
  - Probability of not ending up with attacker: $^{k-m}/_{k}$

Quantitative Analysis of Buckets

- If there are k buckets, n users and $m = n - 1$ attackers:
  - Probability of ending up with honest user: $^1/_k$
  - Probability of not ending up with attacker: $^{k-m}/_k$
    - There is no reason for two attackers to go in one bucket – spread out!

Quantitative Analysis of Buckets

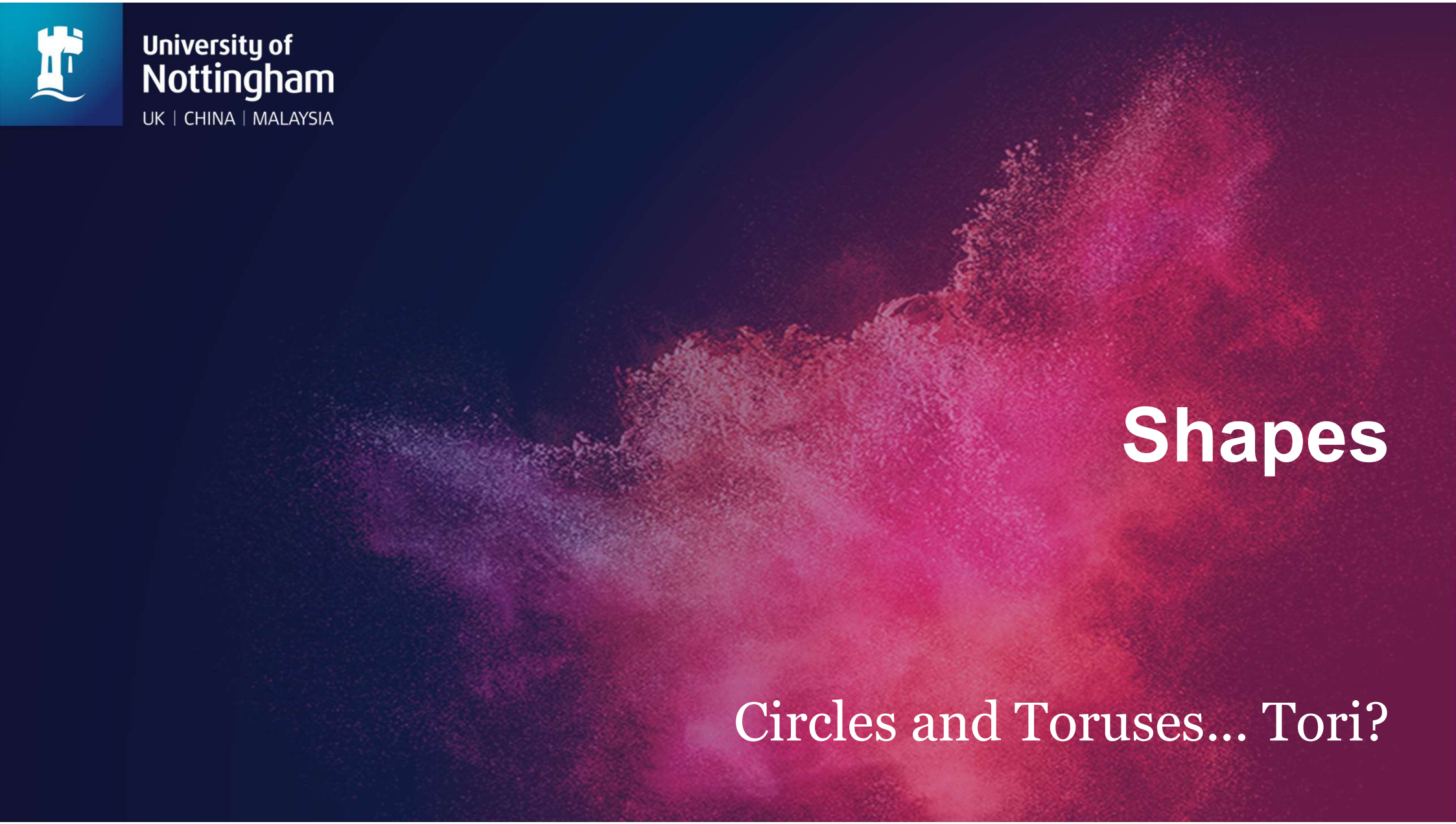
- If there are k buckets, n users and $m = n - 1$ attackers:
  - Probability of ending up with honest user: $^1/_k$
  - Probability of not ending up with attacker: $^{k-m}/_k$
    - There is no reason for two attackers to go in one bucket – spread out!
  - These probabilities are independent, so:
    - Probability of winning is: $^{k-m}/_{k^2}$

Quantitative Analysis of Buckets

- If there are k buckets, n users and $m = n - 1$ attackers:
  - Probability of ending up with honest user: $^1/_k$
  - Probability of not ending up with attacker: $^{k-m}/_k$
    - There is no reason for two attackers to go in one bucket – spread out!
  - These probabilities are independent, so:
    - Probability of winning is: $^{k-m}/_{k^2}$
- Solving for max k: Set $k = 2m$
  - Substituting max k: $\frac{2m-m}{(2m)^2} = \frac{1}{4m} \approx \frac{1}{4n}$
    - This is already reducing to a $O(1)$ advantage. 4 to be precise.
- Wastage: empty buckets & overfull buckets

Recursive Buckets

- Commit to a sequence of buckets instead.
  - If there are 3 or more participants in a bucket, subdivide that group with the next bucket.
- Simplest example:
  - Commit to a bitstring, and pair with longest shared prefix.
  - Attacker will choose all prefixes of length $\log_2(m)$
  - Probability is $1/3m$ if m is a power of 2.
    - And **larger** otherwise, up to $\frac{1}{8/3\ m}$
- It is possible to purposely unbalance the buckets →
  - But empty buckets will remain common ☹


University of
Nottingham
UK | CHINA | MALAYSIA
Shapes
Circles and Toruses... Tori?

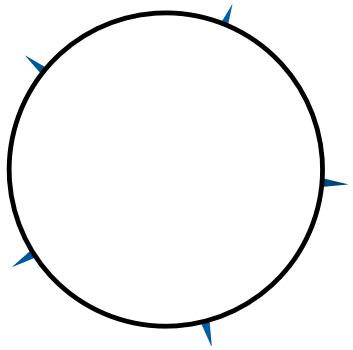
University of Nottingham
UK | CHINA | MALAYSIA

Commit to a Circle

Noli turbare circulos meos!

- Take the real numbers mod 1, a topological circle.
- All parties commit to a value between 0 and 1, and reveal it.
- Pair up users starting with the smallest pairwise distance.
- Optimal attack: space the m $= n - 1$ attackers $\frac{1}{m}$ apart.

- To win:
  - Be in the same interval $\left(\frac{1}{m}\right)$
  - Be closer to each other, than either is to the edge $\left(\frac{1}{3}\right)$
  - Total prob: $\frac{1}{3m}$



Torus

- For recursive buckets, forcing imbalance reduced the factor 3.

- Take a 2d plane of coordinates with Euclidean distance, wrapping around vertically and horizontally.
  - Known to be difficult to find general ways of spacing points equally.

Torus

- For recursive buckets, forcing imbalance reduced the factor 3.
- Take a 2d plane of coordinates with Euclidean distance, wrapping around vertically and horizontally.
  - Known to be difficult to find general ways of spacing points equally.
- For analysis, use numeric methods to find spacings.
  - Practical performance is very close to $\frac{1}{3m}$
  - Higher dimensions do not help ☹


University of
Nottingham
UK | CHINA | MALAYSIA
Hashing
Subtitle

Hashing is the Solution

- W.l.o.g. take h, s.t. $h(a, b) = h(b, a)$ for all a, b.
- All users commit to some values, and then reveal the value.
- For every pair r, s of revealed values, compute $h(r, s)$.
- Pair up users starting with the smallest hash $h(r, s)$.

What's the cheater's advantage?

Quantified Results

- Normalise the hash to [0,1]
- Attackers find a set S of hash values s.t. for every pair $s_i, s_j \in S^2$, $h(s_i, s_j) > 1 - \epsilon$

- With overwhelming probability $h(r, x) < 1 - \epsilon$.
- We win if the honest pair's hash $h(r, t)$ is the smallest hash involving r or t.
- There are m hashes of shape $h(r, s_i)$ and m of shape $h(t, s_i)$.
- So prob. of winning: $^1/_{2m+1} \approx {}^1/_{2n}$

Real Attackers

- For even moderately sized m, a small ϵ is infeasible.
- Imagine you have found a set of hashes for $m-1$ attackers, to add the m^{th} attacker, the probability of a random value s_m satisfying $h(s_m, s_i) < 1 - \epsilon$ for all i, is $\epsilon^{m-1} \approx 0$.
  - Verifying if a random choice works is expensive ($m-1$ hashes)
- If the attacker simply picks random values without regard for pairwise distance, then the protocol is fair: $1/n$
- Todo: full analysis of effort vs pay-off for attacker


University of
Nottingham
UK | CHINA | MALAYSIA
Conclusion
Pretty good

Conclusion

- Typically, biases in RNG protocol give attackers a complete or exponential advantage.
  - We have reduced this to a *constant* advantage in one application.
- Solution is light-weight.
  - Run twice to have prob: $1 - \left(1 - \frac{1}{2n-1}\right)^2 \approx \frac{1}{n}$
  - Far easier to run twice, than to solve a time-lock puzzle!
- Methods to limit bias can have practical applications.

Conclusion

- Typically, biases in RNG protocol give attackers a complete or exponential advantage.
  - We have reduced this to a *constant* advantage in one application.
- Solution is light-weight.
  - Run twice to have prob: $1 - \left(1 - \frac{1}{2n-1}\right)^2 \approx \frac{1}{n}$
  - Far easier to run twice, than to solve a time-lock puzzle!
- Methods to limit bias can have practical applications.
- Somewhat feasible way to get pseudonymity without Sybils.
  - But are we actually afraid of randomness beacons here?
  - Cryptographic solutions may also be suited here.
  - Any stronger motivation/use case?