

Multi-property-preserving Domain Extension Using Polynomial-based Modes of Operation

Jooyoung Lee¹ John P. Steinberger²

¹Electronics and Telecommunications Research Institute

²Institute for Theoretical Computer Science
Tsinghua University

June 3, 2010

Merkle-Damgård transform

Merkle-Damgård transform

- The most popular way to build a cryptographic hash function from a fixed-size compression function
- Preserves collision resistance with an appropriate padding algorithm

If computing collisions becomes somehow feasible for the underlying compression function, then the hash function may fail worse than expected

Generic attacks

- Multicollision attack (Joux, Eurocrypt 2004)
- Long-message second preimage attack (Kelsey and Schneier, Eurocrypt 2005)
- Herding attack (Kelsey and Kohno, Eurocrypt 2006), etc.



Merkle-Damgård transform

Merkle-Damgård transform

- The most popular way to build a cryptographic hash function from a fixed-size compression function
- Preserves collision resistance with an appropriate padding algorithm

If computing collisions becomes somehow feasible for the underlying compression function, then the hash function may fail worse than expected

Generic attacks

- Multicollision attack (Joux, Eurocrypt 2004)
- Long-message second preimage attack (Kelsey and Schneier, Eurocrypt 2005)
- Herding attack (Kelsey and Kohno, Eurocrypt 2006), etc.



Merkle-Damgård transform

Merkle-Damgård transform

- The most popular way to build a cryptographic hash function from a fixed-size compression function
- Preserves collision resistance with an appropriate padding algorithm

If computing collisions becomes somehow feasible for the underlying compression function, then the hash function may fail worse than expected

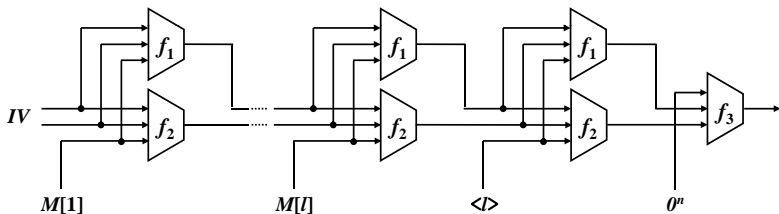
Generic attacks

- Multicollision attack (Joux, Eurocrypt 2004)
- Long-message second preimage attack (Kelsey and Schneier, Eurocrypt 2005)
- Herding attack (Kelsey and Kohno, Eurocrypt 2006), etc.

Wide-pipe strategy

Double-piped mode of operation

- The aforementioned weaknesses can be mitigated by increasing the size of the internal state (Lucks, Asiacrypt 2005)
- The internal state size should be seen as a security parameter of its own right
- Lucks proposed to use a “narrow” compression function in a double-piped mode



Security of double-piped mode of operation

Yasuda analyzed the security of the double-piped mode of operation as a multi-property-preserving domain extension (Eurocrypt 2009)

As a secure message authentication code (MAC)

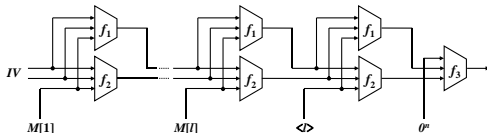
Preserves unforgeability up to $O(2^{5n/6})$ query complexity

As a pseudorandom function

Preserves indistinguishability up to $O(2^n)$ query complexity

As a pseudorandom oracle

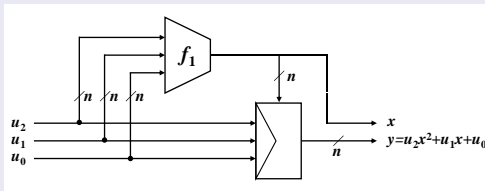
Preserves indistinguishability up to $O(2^n)$ query complexity



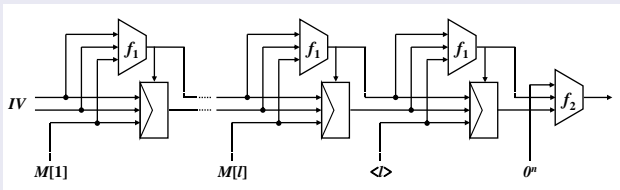
Polynomial-based mode of operation

Efficiency(rate) can be improved by replacing the second primitive by a polynomial

Compression function $\phi[f_1]$



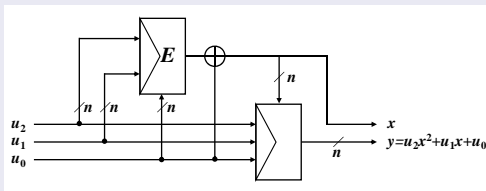
Hash function $H[f_1, f_2]$



Refinements

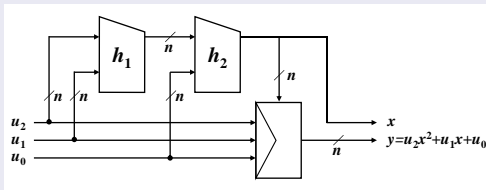
With f_1 replaced by a $2n$ -bit key blockcipher in DM-mode:

Quadratic BC-based function



With f_1 replaced by the cascade of two $2n-n$ bit primitives:

Quadratic cascade function



Security of polynomial-based mode of operation

We analyzed the security of the polynomial-based mode of operation as a multi-property-preserving domain extension

As a secure message authentication code (MAC)

Preserves unforgeability up to $O(2^n/n)$ query complexity

As a pseudorandom function

Preserves indistinguishability up to $O(2^n/n)$ query complexity

As a pseudorandom oracle

Preserves indifferentiability up to $O(2^{2n/3})$ query complexity

Security of polynomial-based mode of operation

We analyzed the security of the polynomial-based mode of operation as a multi-property-preserving domain extension

As a secure message authentication code (MAC)

Preserves unforgeability up to $O(2^n/n)$ query complexity

As a pseudorandom function

Preserves indistinguishability up to $O(2^n/n)$ query complexity

As a pseudorandom oracle

Preserves indistinguishability up to $O(2^{2n/3})$ query complexity

Modular approach for constructing MACs

An and Bellare presented a modular approach for domain extension of a FIL-MAC (Crypto 1999)

Construction of VIL-MAC

- 1 Construct a FIL-WCR using a FIL-MAC.
- 2 Using MD-transform, build a VIL-WCR.
- 3 The composition of the VIL-WCR and a FIL-MAC with an independent key yields a secure VIL-MAC.

The modular approach allows us to focus on the proof of WCR for the polynomial-based compression function



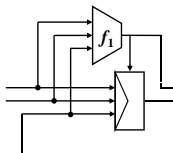
Modular approach for constructing MACs

An and Bellare presented a modular approach for domain extension of a FIL-MAC (Crypto 1999)

Construction of VIL-MAC

- 1 Construct a FIL-WCR using a FIL-MAC.
- 2 Using MD-transform, build a VIL-WCR.
- 3 The composition of the VIL-WCR and a FIL-MAC with an independent key yields a secure VIL-MAC.

The modular approach allows us to focus on the proof of WCR for the polynomial-based compression function



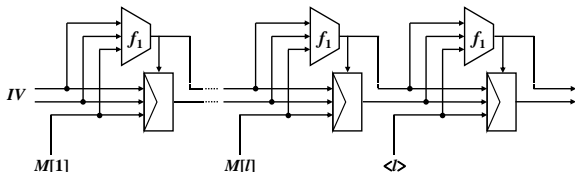
Modular approach for constructing MACs

An and Bellare presented a modular approach for domain extension of a FIL-MAC (Crypto 1999)

Construction of VIL-MAC

- 1 Construct a FIL-WCR using a FIL-MAC.
- 2 Using MD-transform, build a VIL-WCR.
- 3 The composition of the VIL-WCR and a FIL-MAC with an independent key yields a secure VIL-MAC.

The modular approach allows us to focus on the proof of WCR for the polynomial-based compression function



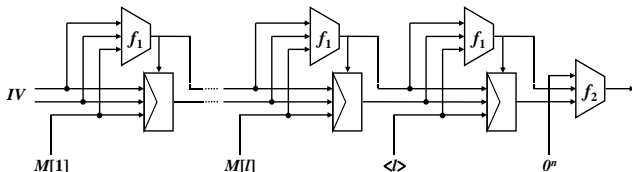
Modular approach for constructing MACs

An and Bellare presented a modular approach for domain extension of a FIL-MAC (Crypto 1999)

Construction of VIL-MAC

- 1 Construct a FIL-WCR using a FIL-MAC.
- 2 Using MD-transform, build a VIL-WCR.
- 3 The composition of the VIL-WCR and a FIL-MAC with an independent key yields a secure VIL-MAC.

The modular approach allows us to focus on the proof of WCR for the polynomial-based compression function



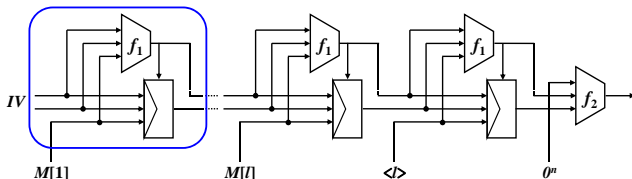
Modular approach for constructing MACs

An and Bellare presented a modular approach for domain extension of a FIL-MAC (Crypto 1999)

Construction of VIL-MAC

- 1 Construct a FIL-WCR using a FIL-MAC.
- 2 Using MD-transform, build a VIL-WCR.
- 3 The composition of the VIL-WCR and a FIL-MAC with an independent key yields a secure VIL-MAC.

The modular approach allows us to focus on the proof of WCR for the polynomial-based compression function



Security definitions

Given a function family $f : \text{Keys} \times \text{Dom} \rightarrow \text{Rng}$,

Unforgeability

Experiment $\text{Exp}_{\mathcal{A}}^{\text{mac}}$

$k \xleftarrow{\$} \text{Keys}$

$(m, \tau) \leftarrow \mathcal{A}^{f_k(\cdot)}$

if $f_k(m) = \tau$, m is "new" then
output 1

else

output 0

Weak collision resistance

Experiment $\text{Exp}_{\mathcal{A}}^{\text{wcr}}$

$k \xleftarrow{\$} \text{Keys}$

$(m, m') \leftarrow \mathcal{A}^{f_k(\cdot)}$

if $f_k(m) = f_k(m'), m \neq m'$ then
output 1

else

output 0

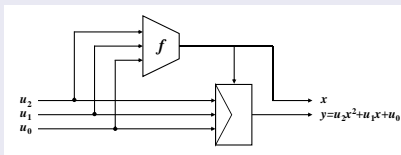
- $\text{Adv}_f^{\text{mac}}(\mathcal{A}) = \Pr [\text{Exp}_{\mathcal{A}}^{\text{mac}} = 1]$
- $\text{Adv}_f^{\text{wcr}}(\mathcal{A}) = \Pr [\text{Exp}_{\mathcal{A}}^{\text{wcr}} = 1]$

WCR of a polynomial-based compression function

Let $\phi[f]$ be the polynomial-based compression function defined by a function family $f : \{0, 1\}^\kappa \times \{0, 1\}^{3n} \rightarrow \{0, 1\}^n$. Then,

$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) \leq 2q(2 + \log q) \mathbf{Adv}_f^{\text{mac}}(t + O(n^2 q^4), q).$$

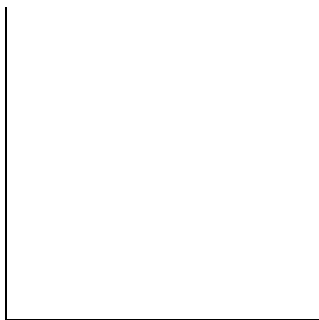
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

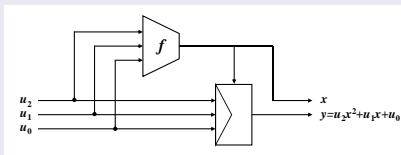
$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines **a quadratic curve**
- f 's response specifies **a point on the curve**

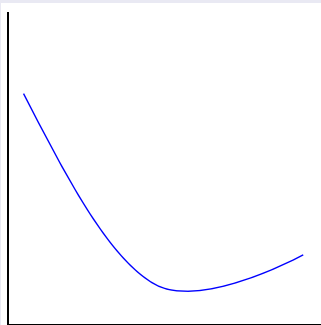
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

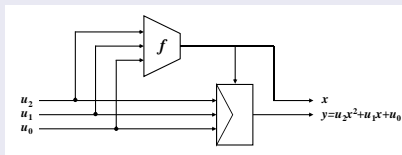
$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines **a quadratic curve**
- f 's response specifies **a point on the curve**

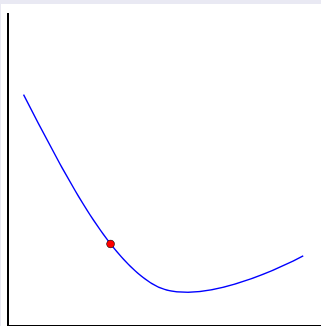
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

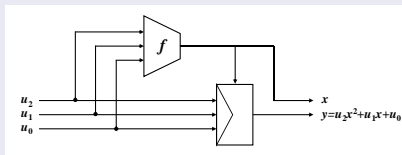
$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines a quadratic curve
- f 's response specifies a point on the curve

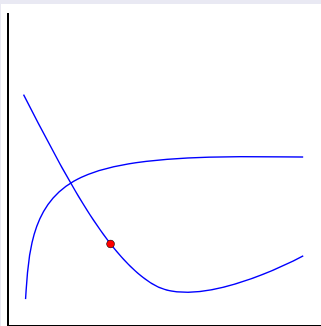
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

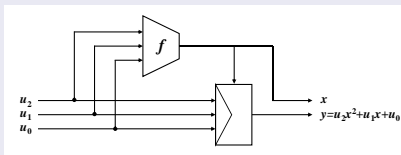
$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines a quadratic curve
- f 's response specifies a point on the curve

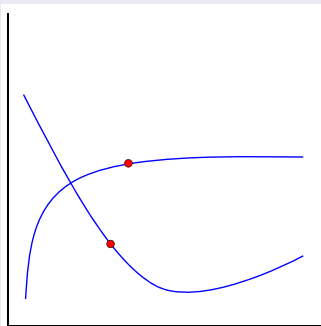
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

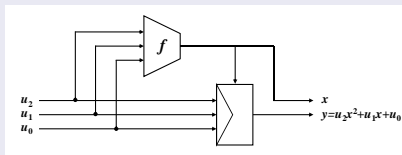
$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines **a quadratic curve**
- f 's response specifies **a point on the curve**

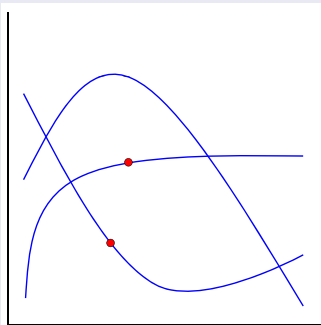
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

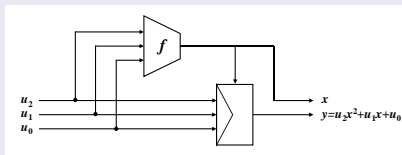
$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines **a quadratic curve**
- f 's response specifies **a point on the curve**

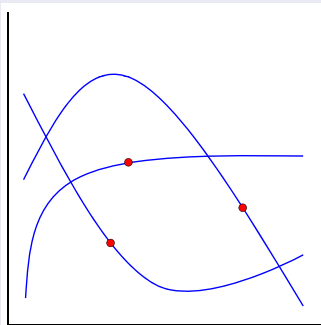
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

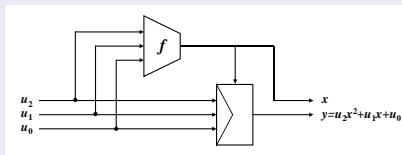
$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines **a quadratic curve**
- f 's response specifies **a point on the curve**

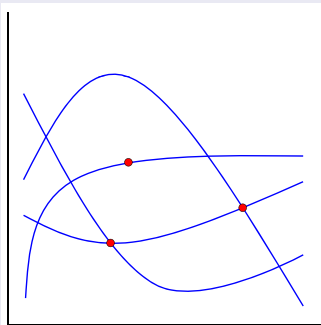
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

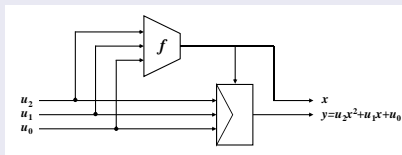
$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines a quadratic curve
- f 's response specifies a point on the curve

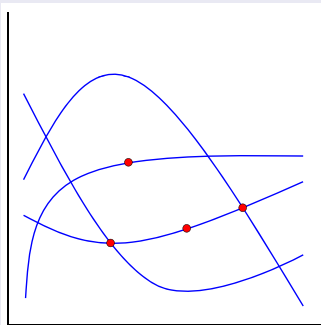
Compression function $\phi[f]$



Let \mathcal{A} be an optimal adversary s.t.

$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(t, q) = \epsilon$$

Interaction btwn \mathcal{A} and f



- Oracle access to either $\phi[f]$ or f is equivalent
- \mathcal{A} 's query determines **a quadratic curve**
- f 's response specifies **a point on the curve**

Case analysis

- We will construct a forger \mathcal{B} of f using \mathcal{A} as a subroutine
- Let $\gamma = \max \gamma_i$, where $\gamma_i = \#$ points already placed on the i -th curve

Case 1

\mathcal{A} finds a collision and $\gamma \leq \log q + 2$

Case 2

$\gamma > \log q + 2$

One of the two cases happens with probability at least $\epsilon/2$

Case analysis

- We will construct a forger \mathcal{B} of f using \mathcal{A} as a subroutine
- Let $\gamma = \max \gamma_i$, where $\gamma_i = \#$ points already placed on the i -th curve

Case 1

\mathcal{A} finds a collision and $\gamma \leq \log q + 2$

Case 2

$\gamma > \log q + 2$

One of the two cases happens with probability at least $\epsilon/2$

Case analysis

- We will construct a forger \mathcal{B} of f using \mathcal{A} as a subroutine
- Let $\gamma = \max \gamma_i$, where $\gamma_i = \#$ points already placed on the i -th curve

Case 1

\mathcal{A} finds a collision and $\gamma \leq \log q + 2$

Case 2

$\gamma > \log q + 2$

One of the two cases happens with probability at least $\epsilon/2$

Case 1: $\Pr [\mathcal{A} \text{ finds a collision} \wedge \gamma \leq \log q + 2] \geq \epsilon/2$

Forger \mathcal{B}

- 1 \mathcal{B} chooses $i \in \{1, \dots, q\}$ uniformly at random
- 2 \mathcal{B} runs \mathcal{A} as a subroutine and faithfully answers the queries made by \mathcal{A} until the $(i-1)$ -th query
- 3 On the i -query, \mathcal{B} presents a forgery by randomly choosing one of the points already placed on the i -th curve

Analysis

- With probability $\geq \frac{\epsilon}{2q}$, \mathcal{B} makes a correct guess of the query that determines a collision
- In this case, \mathcal{B} successfully forges f with probability $\geq \frac{1}{\log q + 2}$
- Therefore, we have $\mathbf{Adv}_f^{\text{mac}}(\mathcal{B}) \geq \frac{\epsilon}{2q(\log q + 2)}$

Case 1: $\Pr[\mathcal{A} \text{ finds a collision} \wedge \gamma \leq \log q + 2] \geq \epsilon/2$

Forger \mathcal{B}

- 1 \mathcal{B} chooses $i \in \{1, \dots, q\}$ uniformly at random
- 2 \mathcal{B} runs \mathcal{A} as a subroutine and faithfully answers the queries made by \mathcal{A} until the $(i-1)$ -th query
- 3 On the i -query, \mathcal{B} presents a forgery by randomly choosing one of the points already placed on the i -th curve

Analysis

- With probability $\geq \frac{\epsilon}{2q}$, \mathcal{B} makes a correct guess of the query that determines a collision
- In this case, \mathcal{B} successfully forges f with probability $\geq \frac{1}{\log q + 2}$
- Therefore, we have $\mathbf{Adv}_f^{\text{mac}}(\mathcal{B}) \geq \frac{\epsilon}{2q(\log q + 2)}$

Case 2: $\Pr[\gamma > \log q + 2] \geq \epsilon/2$

Balls-in-bins game (Dodis and Steinberger, Crypto 2009)

Players: \mathcal{A} and \mathcal{B}

Parameters: q, m_1, m_2, M ($m_1 < m_2$)

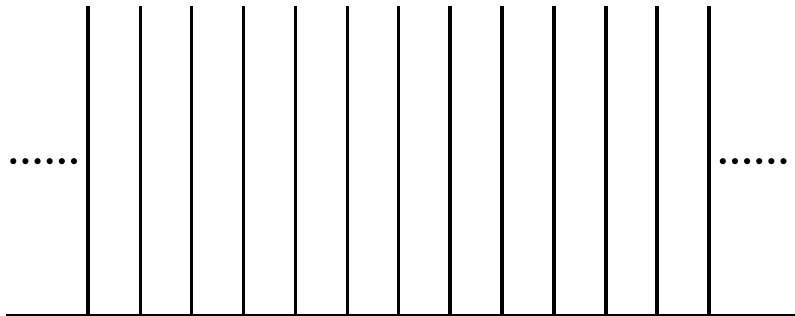
- 1 The game consists of q rounds
- 2 At each round, \mathcal{A} publicly places a set of balls into a set of bins such that
 - 1 balls placed at the same round go into distinct bins,
 - 2 the number of bins containing more than m_1 balls at the end of the game is at most M ,
 - 3 some bin eventually contains more than m_2 balls
- 3 Before each round, \mathcal{B} can secretly “pass” or “guess” a bin that will receive a ball in the next round. \mathcal{B} makes exactly one guess throughout the game
- 4 If \mathcal{B} makes a correct guess, then \mathcal{B} wins. Otherwise, \mathcal{B} loses

Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 1:

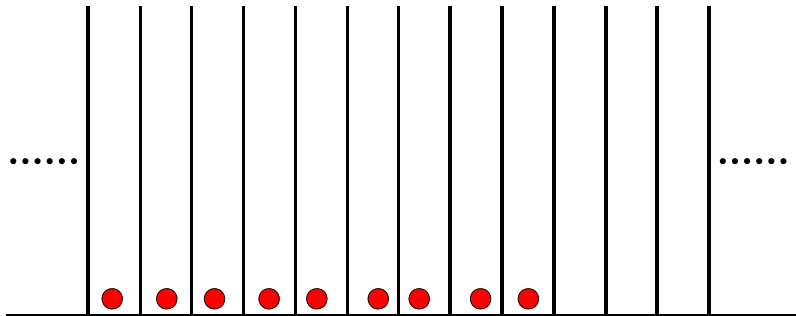


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 1:

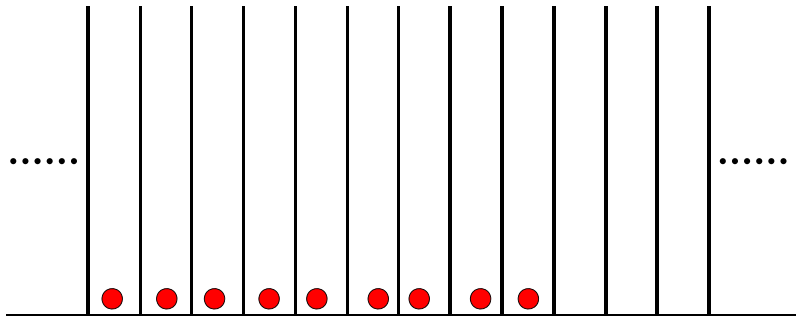


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 2:

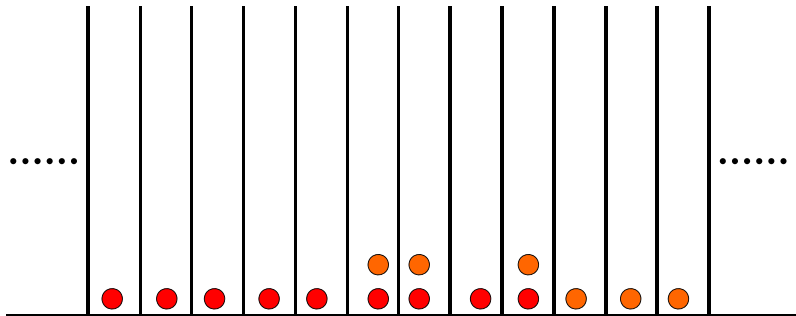


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 2:

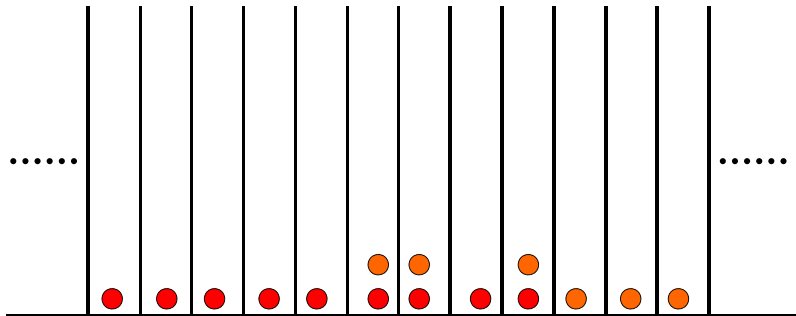


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 3:

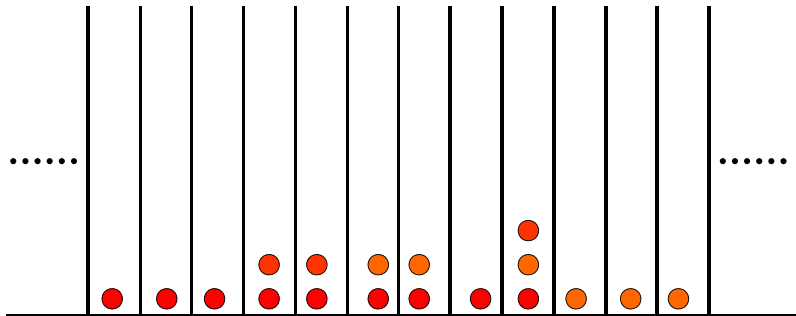


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 3:

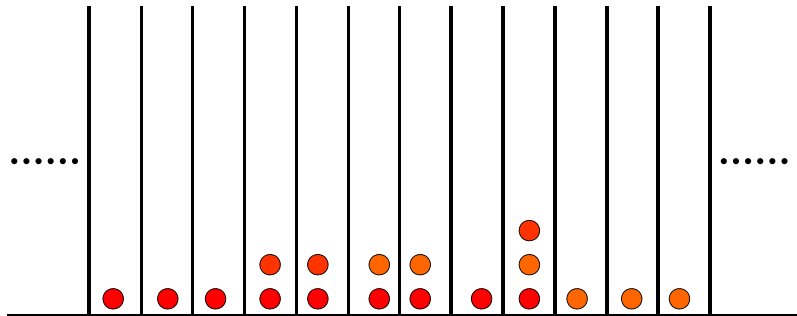


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 4:

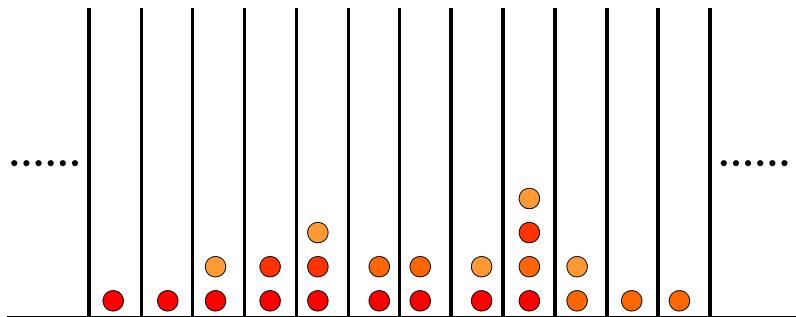


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 4:

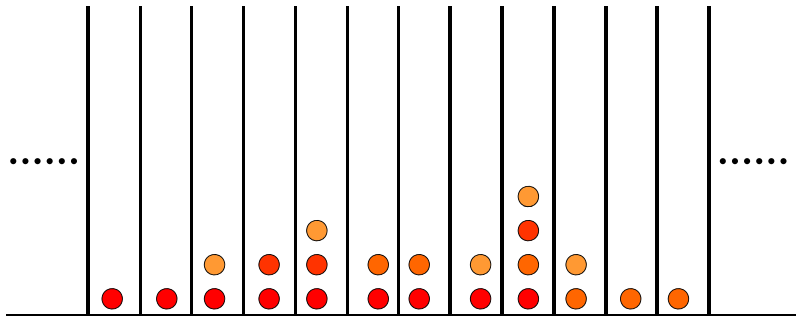


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 5:

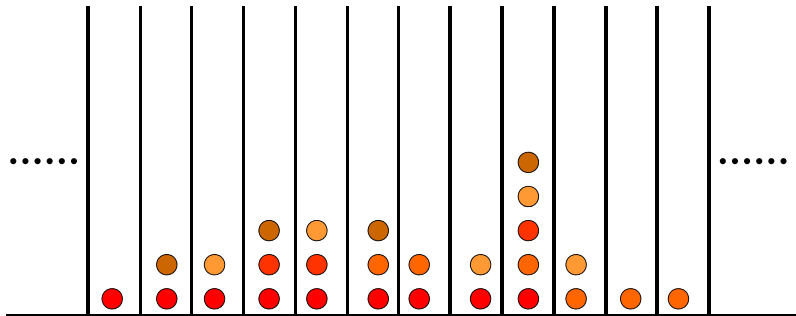


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 5:

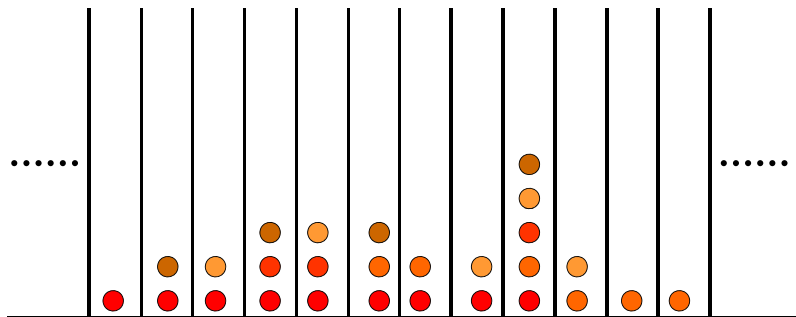


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 6:

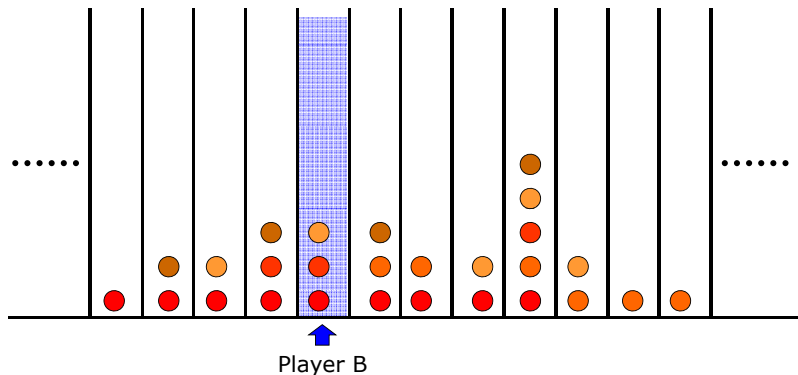


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 6:

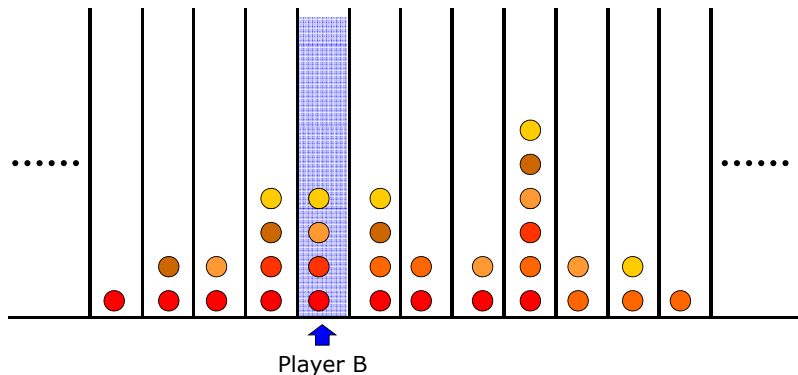


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 6:

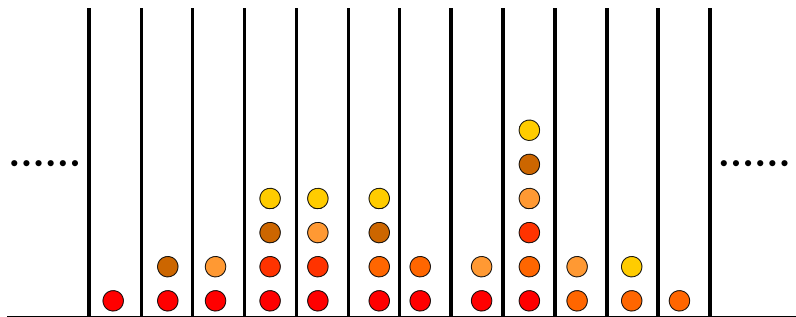


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 7:

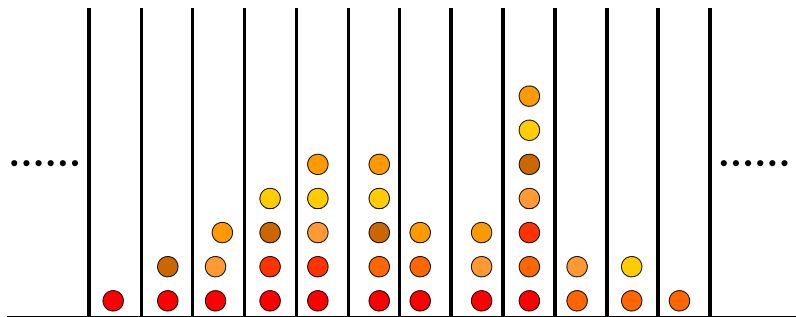


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 7:

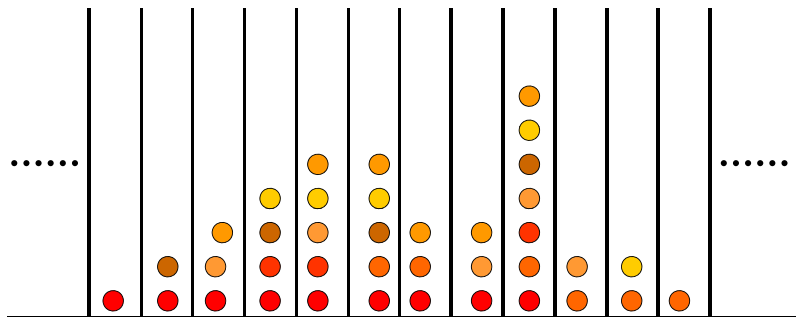


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 8:

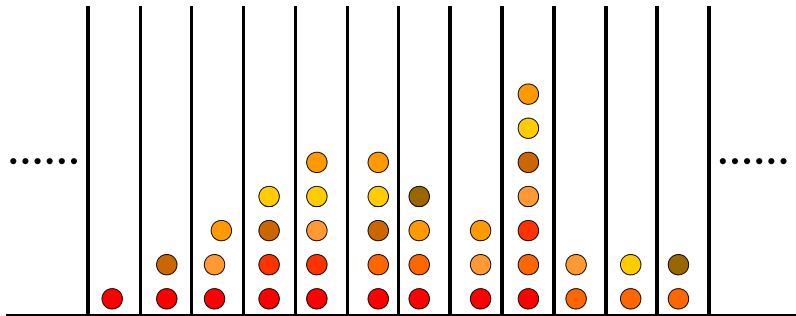


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 8:

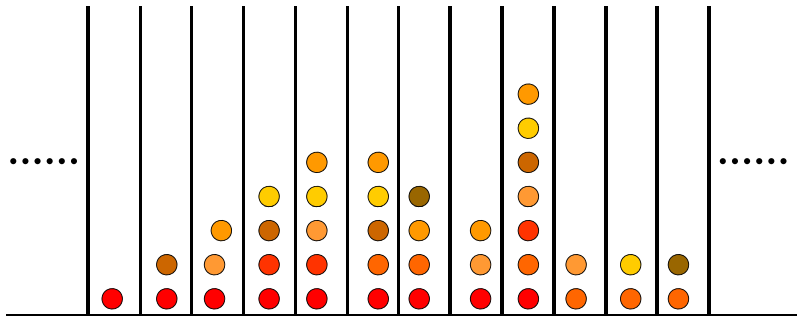


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 9:

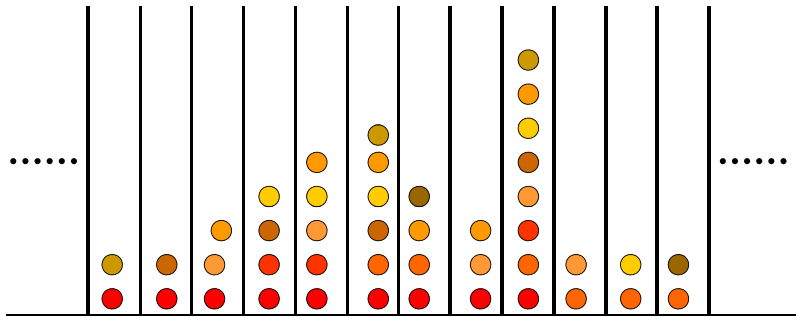


Example: balls-in-bins game

Parameters

$$q = 9, m_1 = 3, m_2 = 7, M = 5$$

Round 9:

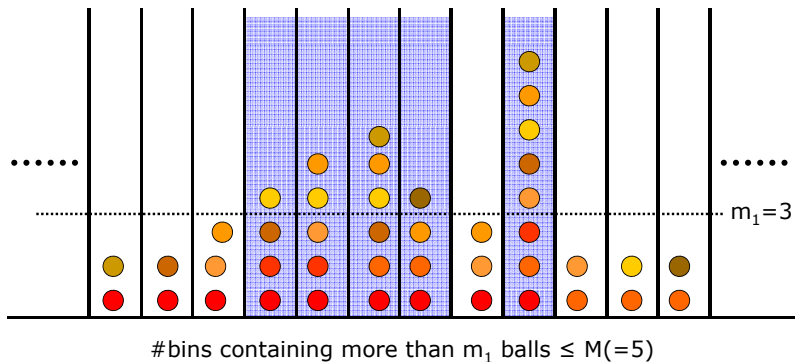


Example: balls-in-bins game

Parameters

$$m_1 = 3, m_2 = 7, M = 5$$

End of the game:

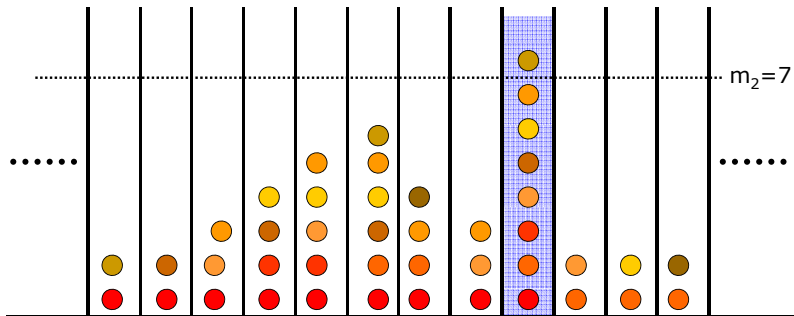


Example: balls-in-bins game

Parameters

$$m_1 = 3, m_2 = 7, M = 5$$

End of the game:



WCF adversary \Rightarrow player \mathcal{A} of a balls-in-bins game

balls-in-bins game: G

- A **ball** is associated with a point in $\mathbb{F}_{2^n}^2$
- A **bin** is a quadratic curve in $\mathbb{F}_{2^n}^2$
- On the i -th query $u[i]$ of \mathcal{A} , the i -th round of the game begins
- Given the i -th point $\phi[f](u[i])$ as a response of f , every quadratic curve containing the point except the i -th curve itself receives a single ball

Parameters

- The game consists of q rounds
- The number of bins that contain more than $m_1 = 2$ balls at the end of the balls-in-bins game is at most $\binom{q}{3} \leq M = q^3$
- At the end of the game, some curve contains more than $m_2 = \log q + 2$ balls

WCF adversary \Rightarrow player \mathcal{A} of a balls-in-bins game

balls-in-bins game: G

- A **ball** is associated with a point in $\mathbb{F}_{2^n}^2$
- A **bin** is a quadratic curve in $\mathbb{F}_{2^n}^2$
- On the i -th query $u[i]$ of \mathcal{A} , the i -th round of the game begins
- Given the i -th point $\phi[f](u[i])$ as a response of f , every quadratic curve containing the point except the i -th curve itself receives a single ball

Parameters

- The game consists of q rounds
- The number of bins that contain more than $m_1 = 2$ balls at the end of the balls-in-bins game is at most $\binom{q}{3} \leq M = q^3$
- At the end of the game, some curve contains more than $m_2 = \log q + 2$ balls

Player \mathcal{B} of game $G \Rightarrow$ forger of f

A player \mathcal{B} of game G with a high probability of winning can be transformed into a successful forger of f

If \mathcal{B} makes a correct guess of the curve before the i -th round, then it can present a forgery of f by computing the intersection of the curve and the i -th curve (Probability $1/2$)

Winning strategy of player \mathcal{B}

Irrespective of \mathcal{A} 's strategy, there exists a strategy for \mathcal{B} to win game G with probability at least $1/q \cdot 1/(qM)^{1/(m_2-m_1)}$

With $m_1 = 2$, $M = q^3$ and $m_2 = \log q + 2$, the player \mathcal{B} can be transformed into a forger such that

$$\text{Adv}_f^{\text{mac}}(\mathcal{B}) \geq \frac{\epsilon}{2} \cdot \frac{1}{2} \cdot \frac{1}{q} \left(\frac{1}{qM} \right)^{1/(m_2-m_1)} = \frac{\epsilon}{64q}$$

Player \mathcal{B} of game $G \Rightarrow$ forger of f

A player \mathcal{B} of game G with a high probability of winning can be transformed into a successful forger of f

If \mathcal{B} makes a correct guess of the curve before the i -th round, then it can present a forgery of f by computing the intersection of the curve and the i -th curve (Probability $1/2$)

Winning strategy of player \mathcal{B}

Irrespective of \mathcal{A} 's strategy, there exists a strategy for \mathcal{B} to win game G with probability at least $1/q \cdot 1/(qM)^{1/(m_2-m_1)}$

With $m_1 = 2$, $M = q^3$ and $m_2 = \log q + 2$, the player \mathcal{B} can be transformed into a forger such that

$$\text{Adv}_f^{\text{mac}}(\mathcal{B}) \geq \frac{\epsilon}{2} \cdot \frac{1}{2} \cdot \frac{1}{q} \left(\frac{1}{qM} \right)^{1/(m_2-m_1)} = \frac{\epsilon}{64q}$$

Player \mathcal{B} of game $G \Rightarrow$ forger of f

A player \mathcal{B} of game G with a high probability of winning can be transformed into a successful forger of f

If \mathcal{B} makes a correct guess of the curve before the i -th round, then it can present a forgery of f by computing the intersection of the curve and the i -th curve (Probability $1/2$)

Winning strategy of player \mathcal{B}

Irrespective of \mathcal{A} 's strategy, there exists a strategy for \mathcal{B} to win game G with probability at least $1/q \cdot 1/(qM)^{1/(m_2-m_1)}$

With $m_1 = 2$, $M = q^3$ and $m_2 = \log q + 2$, the player \mathcal{B} can be transformed into a forger such that

$$\text{Adv}_f^{\text{mac}}(\mathcal{B}) \geq \frac{\epsilon}{2} \cdot \frac{1}{2} \cdot \frac{1}{q} \left(\frac{1}{qM} \right)^{1/(m_2-m_1)} = \frac{\epsilon}{64q}$$

Player \mathcal{B} of game $G \Rightarrow$ forger of f

A player \mathcal{B} of game G with a high probability of winning can be transformed into a successful forger of f

If \mathcal{B} makes a correct guess of the curve before the i -th round, then it can present a forgery of f by computing the intersection of the curve and the i -th curve (Probability $1/2$)

Winning strategy of player \mathcal{B}

Irrespective of \mathcal{A} 's strategy, there exists a strategy for \mathcal{B} to win game G with probability at least $1/q \cdot 1/(qM)^{1/(m_2-m_1)}$

With $m_1 = 2$, $M = q^3$ and $m_2 = \log q + 2$, the player \mathcal{B} can be transformed into a forger such that

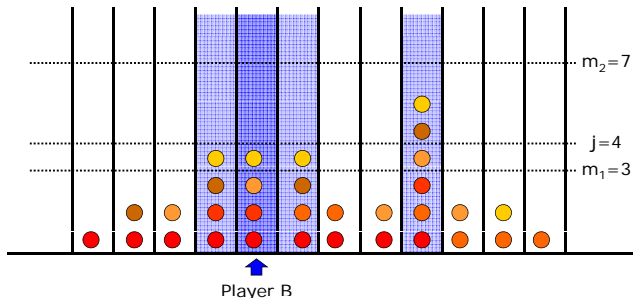
$$\text{Adv}_f^{\text{mac}}(\mathcal{B}) \geq \frac{\epsilon}{2} \cdot \frac{1}{2} \cdot \frac{1}{q} \left(\frac{1}{qM} \right)^{1/(m_2-m_1)} = \frac{\epsilon}{64q}$$

What is the winning strategy of \mathcal{B} ?

\mathcal{B} 's strategy

- 1 Choose a round $i \in \{1, \dots, q\}$ uniformly at random
- 2 Choose a level $j \in \{m_1 + 1, \dots, m_2\}$ uniformly at random
- 3 Before the i -th round of the game, guess a bin uniformly at random from **all bins containing at least j balls already**

Before the 6-th round



Summary

Case 1: $\Pr [\mathcal{A} \text{ finds a collision} \wedge \gamma \leq \log q + 2] \geq \epsilon/2$

There exists a forger \mathcal{B}_1 such that $\mathbf{Adv}_f^{\text{mac}}(\mathcal{B}_1) \geq \frac{\epsilon}{2q(\log q + 2)}$

Case 2: $\Pr [\gamma > \log q + 2] \geq \epsilon/2$

There exists a forger \mathcal{B}_2 such that $\mathbf{Adv}_f^{\text{mac}}(\mathcal{B}_2) \geq \frac{\epsilon}{64q}$

For an optimal forger \mathcal{B} ,

$$\begin{aligned} \mathbf{Adv}_f^{\text{mac}}(\mathcal{B}) &\geq \min \left\{ \frac{1}{2q(\log q + 2)}, \frac{1}{64q} \right\} \times \epsilon \\ &= \frac{1}{2q(\log q + 2)} \mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) \end{aligned}$$

$$\mathbf{Adv}_{\phi[f]}^{\text{wcr}}(\mathcal{A}) \leq 2q(\log q + 2) \mathbf{Adv}_f^{\text{mac}}(\mathcal{B})$$

Thank You