On the Security of Tandem-DM

Ewan Fleischmann, Michael Gorski, Stefan Lucks

Bauhaus-University Weimar

February 23, 2009

Approaches to building a cryptographic hash function

- From Scratch (MD4, MD5, SHA-0/1, SHA-256/512, RIPEMD, ...)
- From a blockcipher (MMO, DM, MDC-2/4, Tandem-DM, Abreast-DM, ...)
- From number-theoretic primitives or hard problems (lattices, modular arithmetic, ...)

Approaches to building a cryptographic hash function

- From Scratch (MD4, MD5, SHA-0/1, SHA-256/512, RIPEMD, ...)
- From a blockcipher (MMO, DM, MDC-2/4, Tandem-DM, Abreast-DM, ...)
- From number-theoretic primitives or hard problems (lattices, modular arithmetic, ...)

Approaches to building a cryptographic hash function

- From Scratch (MD4, MD5, SHA-0/1, SHA-256/512, RIPEMD, ...)
- From a blockcipher (MMO, DM, MDC-2/4, Tandem-DM, Abreast-DM, ...)
- From number-theoretic primitives or hard problems (lattices, modular arithmetic, ...)

Approaches to building a cryptographic hash function

- From Scratch (MD4, MD5, SHA-0/1, SHA-256/512, RIPEMD, ...)
- From a blockcipher (MMO, DM, MDC-2/4, Tandem-DM, Abreast-DM, ...)
- From number-theoretic primitives or hard problems (lattices, modular arithmetic, ...)

Blockcipher Based Hashing - Why?

- Several attacks on MD4-type functions in recent years (MD4/5, SHA family, RIPEMED, ...)
- Only one primitve for encryption and hashing
- Low cost hardware

Blockcipher Based Hashing - Why?

- Several attacks on MD4-type functions in recent years (MD4/5, SHA family, RIPEMED, ...)
- Only one primitve for encryption and hashing
- Low cost hardware

Blockcipher Based Hashing - Why?

- Several attacks on MD4-type functions in recent years (MD4/5, SHA family, RIPEMED, ...)
- Only one primitve for encryption and hashing
- Low cost hardware

Blockcipher Based Hashing - Why not?

- Usually slower than dedicated hash function
- Weaknesses not relevant for encryption (e.g. DES weak keys)
- Output length too short (e.g. 128 bits for AES)
- $\Longrightarrow$ double block length constructions needed (e.g. hash output size of 256 bits for AES)

Blockcipher Based Hashing - Why not?

- Usually slower than dedicated hash function
- Weaknesses not relevant for encryption (e.g. DES weak keys)
- Output length too short (e.g. 128 bits for AES)
- $\Longrightarrow$ double block length constructions needed (e.g. hash output size of 256 bits for AES)

Blockcipher Based Hashing - Why not?

- Usually slower than dedicated hash function
- Weaknesses not relevant for encryption (e.g. DES weak keys)
- Output length too short (e.g. 128 bits for AES)
- $\Longrightarrow$ double block length constructions needed (e.g. hash output size of 256 bits for AES)

Blockcipher Based Hashing - Why not?

- Usually slower than dedicated hash function
- Weaknesses not relevant for encryption (e.g. DES weak keys)
- Output length too short (e.g. 128 bits for AES)
- $\Longrightarrow$ double block length constructions needed (e.g. hash output size of 256 bits for AES)

Blockcipher Based Hashing - The Goal

- 'Secure' (ideal cipher model)
- e.g. birthday type collision resistance
- Long hash output (e.g. >> 128 bits = blocksize)
- Efficient: efficiency = $\dfrac{\text{size of message input}}{\text{number of blockcipher calls needed to process this input}}$

Blockcipher Based Hashing - The Goal

- 'Secure' (ideal cipher model)
- e.g. birthday type collision resistance
- Long hash output (e.g. >> 128 bits = blocksize)
- Efficient: efficiency = $\frac{\text{size of message input}}{\text{number of blockcipher calls needed to process this input}}$

Blockcipher Based Hashing - The Goal

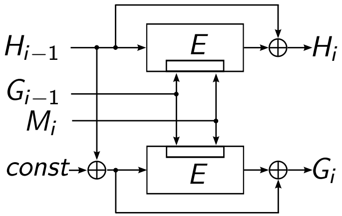
- 'Secure' (ideal cipher model)
- e.g. birthday type collision resistance
- Long hash output (e.g. $>>$ 128 bits $=$ blocksize)
- Efficient: efficiency $= \dfrac{\text{size of message input}}{\text{number of blockcipher calls needed to process this input}}$

Blockcipher Based Hashing - The Goal

- 'Secure' (ideal cipher model)
- e.g. birthday type collision resistance
- Long hash output (e.g. $>>$ 128 bits $=$ blocksize)
- Efficient: efficiency $=$

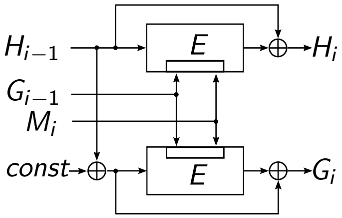
$$\frac{\text{size of message input}}{\text{number of blockcipher calls needed to process this input}}$$

Example: Hirose's FSE'06 proposal



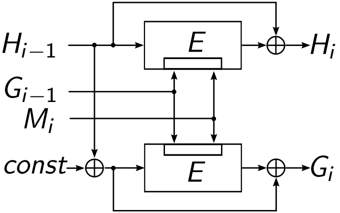
- Rate 1/2, Output size: 2n (i.e. AES-256 256 bits)
- Collision Resistance: $> 2^{124.5}$ for CF
- $(n, 2n)$-blockchiffre, n-bit cipher/plaintext, 2n-bit key

Example: Hirose's FSE'06 proposal



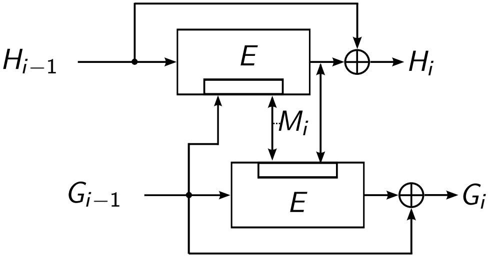
- Rate 1/2, Output size: 2n (i.e. AES-256 256 bits)
- Collision Resistance: $> 2^{124.5}$ for CF
- $(n, 2n)$-blockchiffre, n-bit cipher/plaintext, 2n-bit key

Example: Hirose's FSE'06 proposal



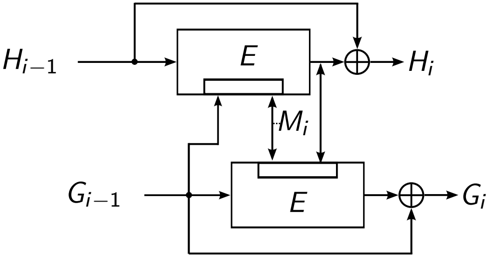
- Rate 1/2, Output size: 2n (i.e. AES-256 256 bits)
- Collision Resistance: $> 2^{124.5}$ for CF
- $(n, 2n)$-blockchiffre, n-bit cipher/plaintext, 2n-bit key

Tandem-DM - a DBL hash function



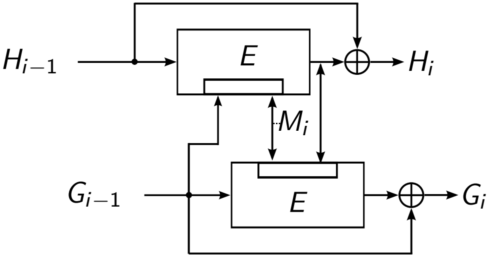
- Rate 1/2, Output: $2n$ (i.e. AES-256: $2n = 256$-bit)
- Proof of Collision Resistance: this presentation/paper
- $(n, 2n)$-blockchiffre, n-bit cipher/plaintext, 2n-bit key

Tandem-DM - a DBL hash function



- Rate 1/2, Output: $2n$ (i.e. AES-256: $2n = 256$-bit)
- Proof of Collision Resistance: this presentation/paper
- $(n, 2n)$-blockchiffre, n-bit cipher/plaintext, 2n-bit key

Tandem-DM - a DBL hash function



- Rate 1/2, Output: $2n$ (i.e. AES-256: $2n = 256$-bit)
- Proof of Collision Resistance: this presentation/paper
- $(n, 2n)$-blockchiffre, n-bit cipher/plaintext, 2n-bit key

Security Bound

Theorem (Bound for Collision Resistance)

Let F be the Tandem-DM compression function and n, q be natural numbers with $q < 2^n$. Let $N' = 2^n - q$ and let α be any positive number with $eq/N' \leq \alpha$ and $\tau = \alpha N'/q$ (and e^x being the exponential function). Then

$$\mathbf{Adv}_F^{\mathrm{COLL}}(q) \leq q 2^n e^{q\tau(1-\ln\tau)/N'} + 4q\alpha/N' + 6q/(N')^2 + 2q/(N')^3.$$

Security Bound

Corollary/Conjecture

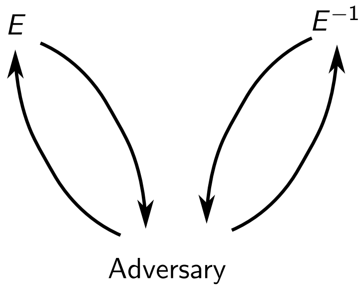
For the compression function Tandem-DM, instantiated with AES-256, any adversary asking less than $2^{120.4}$ (backward or forward) oracle queries cannot find a collision with probability greater than $1/2$. In this case, $\alpha = 24.0$.

Theorem (Bound for Preimage Resistance)

Let $F := F^{TDM}$ be the Tandem-DM compression function. For every $N' = 2^n - q$ and $q > 1$

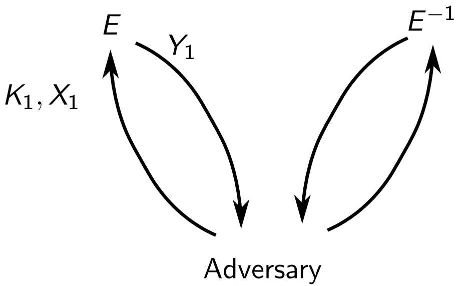
$$\mathbf{Adv}_F^{Inv}(q) \leq 2q/(N')^2.$$

Model for the proof (1)



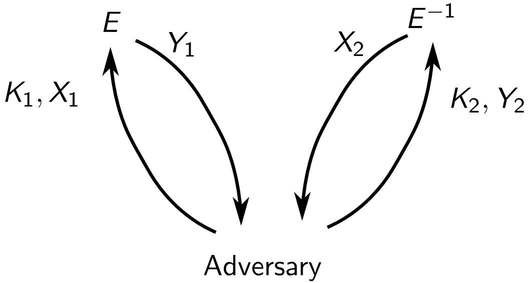
Query History $\mathcal{Q}$: $\{\}$

Model for the proof (1)



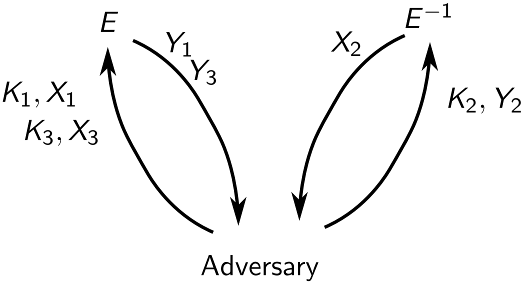
<u>Query History $\mathcal{Q}$:</u> $\{(X_1, K_1, Y_1)\}$

Model for the proof (1)



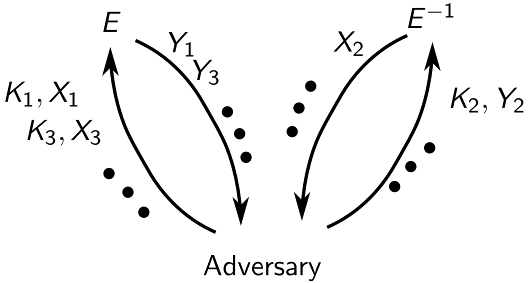
Query History $\mathcal{Q}$: $\{(X_1, K_1, Y_1), (X_2, K_2, Y_2)\}$

Model for the proof (1)



Query History $\mathcal{Q}$: $\{(X_1, K_1, Y_1), (X_2, K_2, Y_2), (X_3, K_3, Y_3)\}$

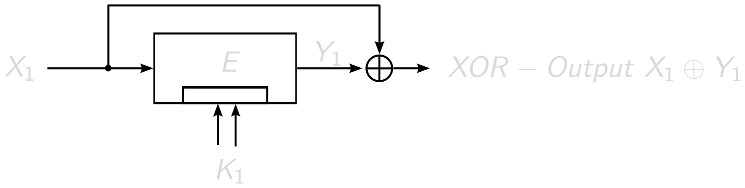
Model for the proof (1)



Query History $\mathcal{Q}$: $\{(X_1, K_1, Y_1), (X_2, K_2, Y_2), (X_3, K_3, Y_3), \ldots\}$

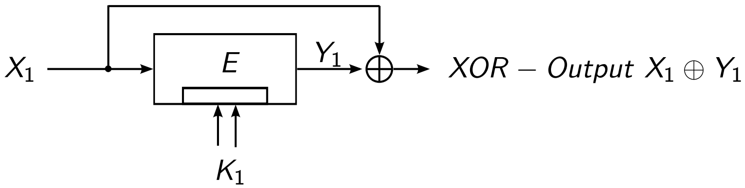
Model for the proof (2)

Query History $\mathcal{Q}$: $\{(X_1, K_1, Y_1), (X_2, K_2, Y_2), (X_3, K_3, Y_3), \ldots\}$



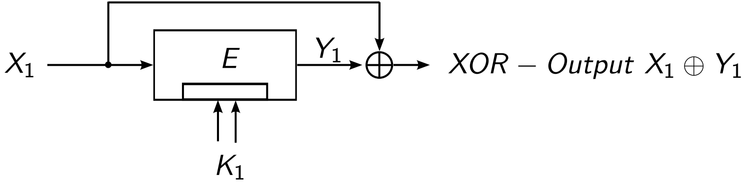
Model for the proof (2)

Query History $\mathcal{Q}$: $\{\underline{(X_1, K_1, Y_1)}, (X_2, K_2, Y_2), (X_3, K_3, Y_3), \ldots\}$



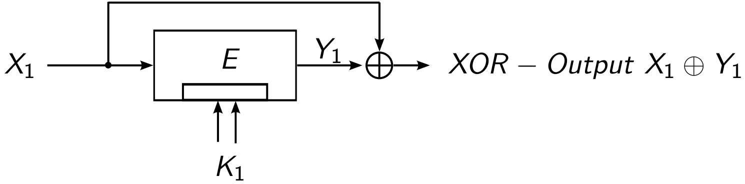
Model for the proof (2)

Query History $\mathcal{Q}$: $\{(X_1, K_1, Y_1), (X_2, K_2, Y_2), (X_3, K_3, Y_3), \ldots\}$



Model for the proof (2)

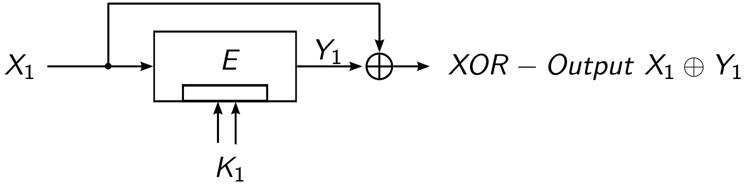
Query History $\mathcal{Q}$: $\{(X_1, K_1, Y_1), (X_2, K_2, Y_2), (X_3, K_3, Y_3), \ldots\}$



1. Adversary A wins if queries can be assembled to hash two distinct colliding words
2. Advantage(A) is the probability of A winning
3. $Adv(q)$ is the max of Advantage(A) taken over all adversaries making at most q queries.

Model for the proof (2)

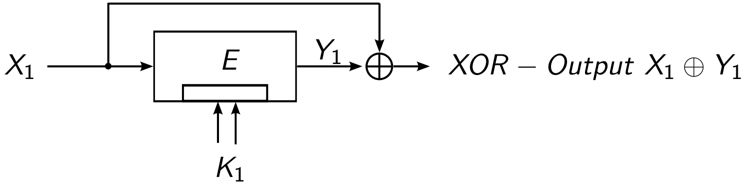
Query History $\mathcal{Q}$: $\{(X_1, K_1, Y_1), (X_2, K_2, Y_2), (X_3, K_3, Y_3), \ldots\}$



1. Adversary A wins if queries can be assembled to hash two distinct colliding words
2. Advantage(A) is the probability of A winning
3. $Adv(q)$ is the max of Advantage(A) taken over all adversaries making at most q queries.

Model for the proof (2)

Query History $\mathcal{Q}$: $\{(X_1, K_1, Y_1), (X_2, K_2, Y_2), (X_3, K_3, Y_3), \ldots\}$

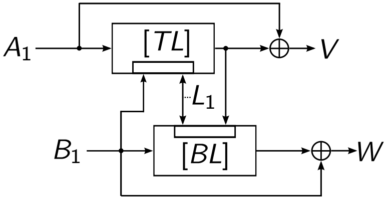


1. Adversary A wins if queries can be assembled to hash two distinct colliding words
2. Advantage(A) is the probability of A winning
3. $Adv(q)$ is the max of Advantage(A) taken over all adversaries making at most q queries.

Main Idea

Main Idea

Upper bound the probability of the adversary making a query that can be used as the *final* query to complete a collision.



Analysis Details

1. We examine the adversarys queries one at a time as they come in
2. The latest query made by the adversary is 'successful' if the adversary can use it to build a collision
3. We upper bound the probability of a query being successful, and multiply this probability by q

Analysis Details

1. We examine the adversarys queries one at a time as they come in
2. The latest query made by the adversary is 'successful' if the adversary can use it to build a collision
3. We upper bound the probability of a query being successful, and multiply this probability by q

Analysis Details

1. We examine the adversarys queries one at a time as they come in
2. The latest query made by the adversary is 'successful' if the adversary can use it to build a collision
3. We upper bound the probability of a query being successful, and multiply this probability by q

Some difficulties

1. A query can be used in many different ways to complete a collision (in different positions of the diagram, or several different times in a diagram)
2. All cases require separate analysis
3. The probability of success of a query will depend on the previous query history $\mathcal{Q}$

Some difficulties

1. A query can be used in many different ways to complete a collision (in different positions of the diagram, or several different times in a diagram)
2. All cases require separate analysis
3. The probability of success of a query will depend on the previous query history Q

Some difficulties

1. A query can be used in many different ways to complete a collision (in different positions of the diagram, or several different times in a diagram)
2. All cases require separate analysis
3. The probability of success of a query will depend on the previous query history $\mathcal{Q}$

Proof Overview

1. Exhibit predicates $\textsc{Lucky}(\mathcal{Q})$, $\textsc{Win1}(\mathcal{Q})$, $\textsc{Win2}(\mathcal{Q})$ and $\textsc{Win3}(\mathcal{Q})$ such that
2. $\textsc{Coll}^{TDM}(\mathcal{Q}) \Rightarrow \textsc{Lucky}(\mathcal{Q}) \vee \textsc{Win1}(\mathcal{Q}) \vee \textsc{Win2}(\mathcal{Q}) \vee \textsc{Win3}(\mathcal{Q})$
3. Upper bound separately the probabilities $\Pr[\textsc{Lucky}(\mathcal{Q})]$, $\Pr[\textsc{Win1}(\mathcal{Q})]$, $\Pr[\textsc{Win2}(\mathcal{Q})]$ and $\Pr[\textsc{Win3}(\mathcal{Q})]$
4. Then $\Pr[\textsc{Coll}(\mathcal{Q})] \leq \Pr[\textsc{Lucky}(\mathcal{Q})] + \Pr[\textsc{Win1}(\mathcal{Q})] + \Pr[\textsc{Win2}(\mathcal{Q})] + \Pr[\textsc{Win3}(\mathcal{Q})]$.

Proof Overview

1. Exhibit predicates $\textsc{Lucky}(\mathcal{Q})$, $\textsc{Win1}(\mathcal{Q})$, $\textsc{Win2}(\mathcal{Q})$ and $\textsc{Win3}(\mathcal{Q})$ such that
2. $\textsc{Coll}^{TDM}(\mathcal{Q}) \Rightarrow \textsc{Lucky}(\mathcal{Q}) \vee \textsc{Win1}(\mathcal{Q}) \vee \textsc{Win2}(\mathcal{Q}) \vee \textsc{Win3}(\mathcal{Q})$
3. Upper bound separately the probabilities $\Pr[\textsc{Lucky}(\mathcal{Q})]$, $\Pr[\textsc{Win1}(\mathcal{Q})]$, $\Pr[\textsc{Win2}(\mathcal{Q})]$ and $\Pr[\textsc{Win3}(\mathcal{Q})]$
4. Then $\Pr[\textsc{Coll}(\mathcal{Q})] \leq \Pr[\textsc{Lucky}(\mathcal{Q})]+\Pr[\textsc{Win1}(\mathcal{Q})]+\Pr[\textsc{Win2}(\mathcal{Q})]+\Pr[\textsc{Win3}(\mathcal{Q})]$.

Proof Overview

1. Exhibit predicates $\textsc{Lucky}(\mathcal{Q})$, $\textsc{Win1}(\mathcal{Q})$, $\textsc{Win2}(\mathcal{Q})$ and $\textsc{Win3}(\mathcal{Q})$ such that
2. $\textsc{Coll}^{TDM}(\mathcal{Q}) \Rightarrow \textsc{Lucky}(\mathcal{Q}) \vee \textsc{Win1}(\mathcal{Q}) \vee \textsc{Win2}(\mathcal{Q}) \vee \textsc{Win3}(\mathcal{Q})$
3. Upper bound separately the probabilities $\Pr[\textsc{Lucky}(\mathcal{Q})]$, $\Pr[\textsc{Win1}(\mathcal{Q})]$, $\Pr[\textsc{Win2}(\mathcal{Q})]$ and $\Pr[\textsc{Win3}(\mathcal{Q})]$
4. Then $\Pr[\textsc{Coll}(\mathcal{Q})] \leq \Pr[\textsc{Lucky}(\mathcal{Q})] + \Pr[\textsc{Win1}(\mathcal{Q})] + \Pr[\textsc{Win2}(\mathcal{Q})] + \Pr[\textsc{Win3}(\mathcal{Q})]$.

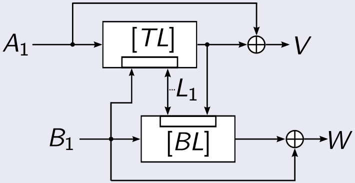
Proof Overview

1. Exhibit predicates $\textsc{Lucky}(\mathcal{Q})$, $\textsc{Win1}(\mathcal{Q})$, $\textsc{Win2}(\mathcal{Q})$ and $\textsc{Win3}(\mathcal{Q})$ such that
2. $\textsc{Coll}^{TDM}(\mathcal{Q}) \Rightarrow$ $\textsc{Lucky}(\mathcal{Q}) \vee \textsc{Win1}(\mathcal{Q}) \vee \textsc{Win2}(\mathcal{Q}) \vee \textsc{Win3}(\mathcal{Q})$
3. Upper bound separately the probabilities $\Pr[\textsc{Lucky}(\mathcal{Q})]$, $\Pr[\textsc{Win1}(\mathcal{Q})]$, $\Pr[\textsc{Win2}(\mathcal{Q})]$ and $\Pr[\textsc{Win3}(\mathcal{Q})]$
4. Then $\Pr[\textsc{Coll}(\mathcal{Q})] \leq$ $\Pr[\textsc{Lucky}(\mathcal{Q})]+\Pr[\textsc{Win1}(\mathcal{Q})]+\Pr[\textsc{Win2}(\mathcal{Q})]+\Pr[\textsc{Win3}(\mathcal{Q})]$.

Example Case: $Win1(\mathcal{Q}) = \neg\textsc{Lucky}(\mathcal{Q}) \wedge Fit1(\mathcal{Q})$

Fit$1(\mathcal{Q})$**:** The last query is used only once in position TL. Note that this is equal to the case where the last query is used only once in position TR.

$\textsc{Fit}1a(\mathcal{Q})$	*all queries used in the collision are pairwise different,*
$\textsc{Fit}1b(\mathcal{Q})$	$\mathrm{BL} = \mathrm{TR}$ *and* BR *is different to* TL, BL, TR,
$\textsc{Fit}1c(\mathcal{Q})$	$\mathrm{BL} = \mathrm{BR}$ *and* TR *is different to* TL, BL, BR,
$\textsc{Fit}1d(\mathcal{Q})$	$\mathrm{TR} = \mathrm{BR}$ *and* BL *is different to* TL, TR, BR,



Concluding Remarks

1. Only two rate 1/2 DBL compression functions with birthday type collision resistance known: Hirose FSE'06 and Tandem-DM
2. Tandem-DM (Eurocrypt'92) took $>$ 15 years for a security proof
3. Still missing tight proofs for e.g. MDC-2, MDC-4, ...
4. essentially no generic results known in this field
5. needs to be a lot more research done

Concluding Remarks

1. Only two rate 1/2 DBL compression functions with birthday type collision resistance known: Hirose FSE'06 and Tandem-DM
2. Tandem-DM (Eurocrypt'92) took $>$ 15 years for a security proof
3. Still missing tight proofs for e.g. MDC-2, MDC-4, ...
4. essentially no generic results known in this field
5. needs to be a lot more research done

Concluding Remarks

1. Only two rate 1/2 DBL compression functions with birthday type collision resistance known: Hirose FSE'06 and Tandem-DM
2. Tandem-DM (Eurocrypt'92) took $>$ 15 years for a security proof
3. Still missing tight proofs for e.g. MDC-2, MDC-4, ...
4. essentially no generic results known in this field
5. needs to be a lot more research done

Concluding Remarks

1. Only two rate 1/2 DBL compression functions with birthday type collision resistance known: Hirose FSE'06 and Tandem-DM
2. Tandem-DM (Eurocrypt'92) took $>$ 15 years for a security proof
3. Still missing tight proofs for e.g. MDC-2, MDC-4, ...
4. essentially no generic results known in this field
5. needs to be a lot more research done

Concluding Remarks

1. Only two rate 1/2 DBL compression functions with birthday type collision resistance known: Hirose FSE'06 and Tandem-DM
2. Tandem-DM (Eurocrypt'92) took $>$ 15 years for a security proof
3. Still missing tight proofs for e.g. MDC-2, MDC-4, ...
4. essentially no generic results known in this field
5. needs to be a lot more research done