Understanding Multi-Key Security Degradation

Atul Luykx[1,2] Bart Mennink[1] Kenny Paterson[3]

[1]ESAT/COSIC, KU Leuven and iMinds, Belgium

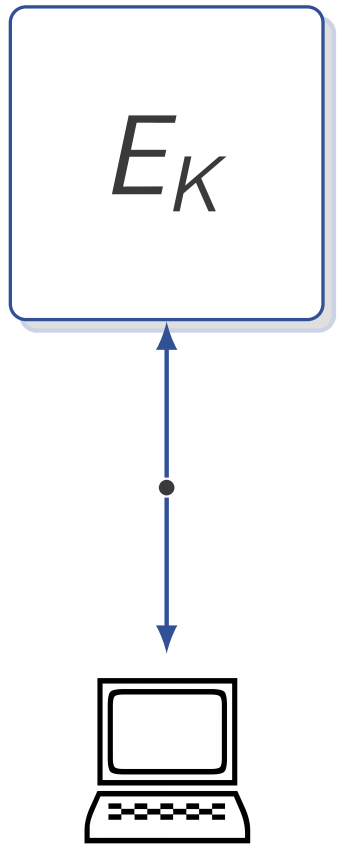
[2]Computer Science Department, UC Davis, USA

[3]Information Security Group, Royal Holloway, University of London, UK

September 30, 2016

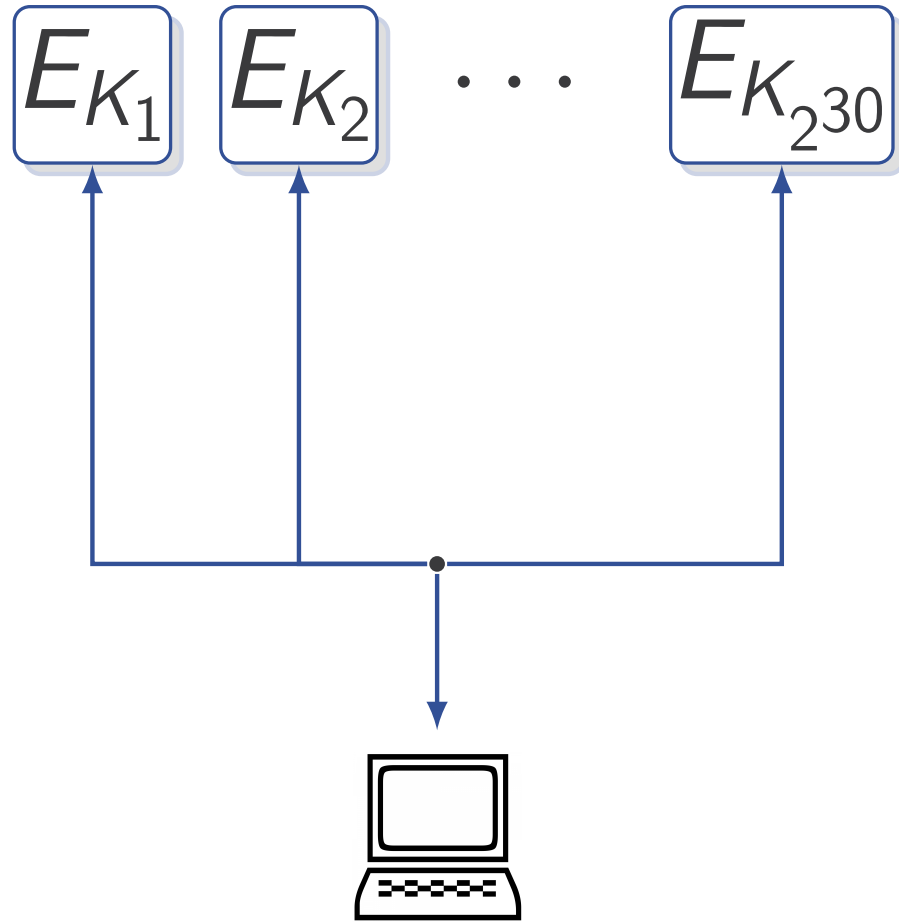
Single-Key vs. Multi-Key

1. Single-key setting usually analyzed



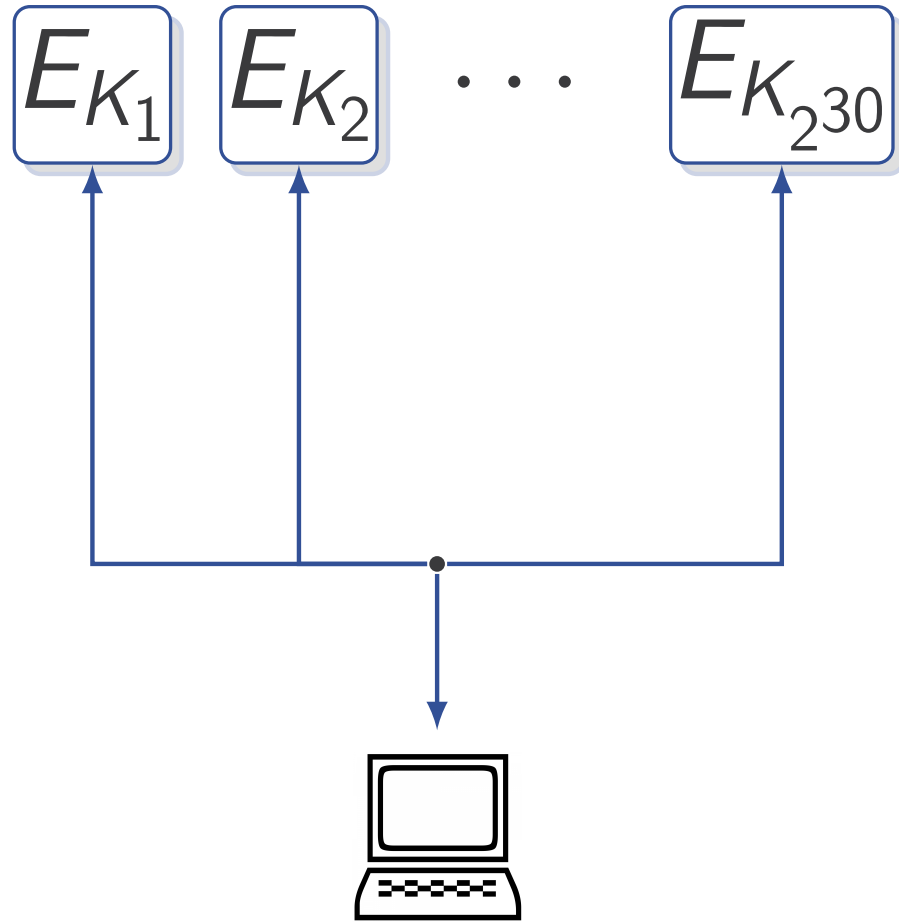
Single-Key vs. Multi-Key

1. Single-key setting usually analyzed
2. Multi-key setting is practically important



Single-Key vs. Multi-Key

1. Single-key setting usually analyzed
2. Multi-key setting is practically important
3. Example: AES-GCM used in TLS, hundreds of millions of keys used



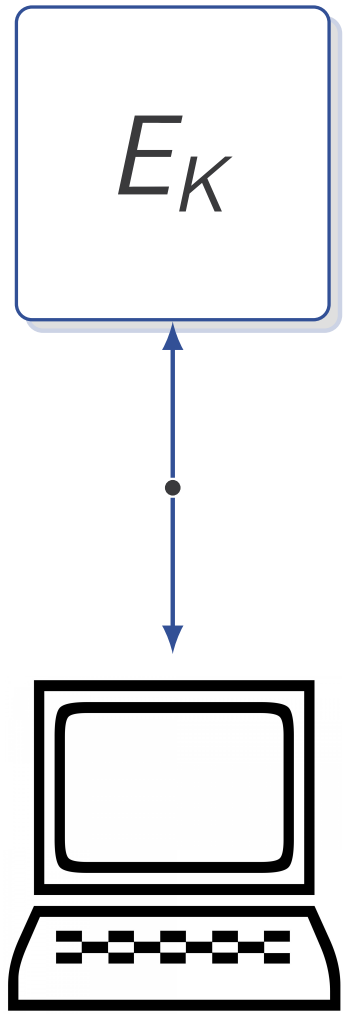
Folklore Result and Multi-Key Degradation

Multi-Key Success Probability

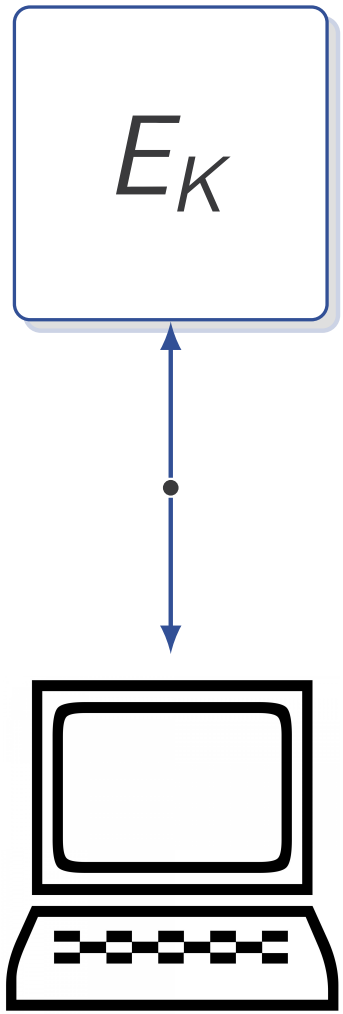
$\leq$

$u\times$ Single-Key Success Probability

Example : Block Ciphers

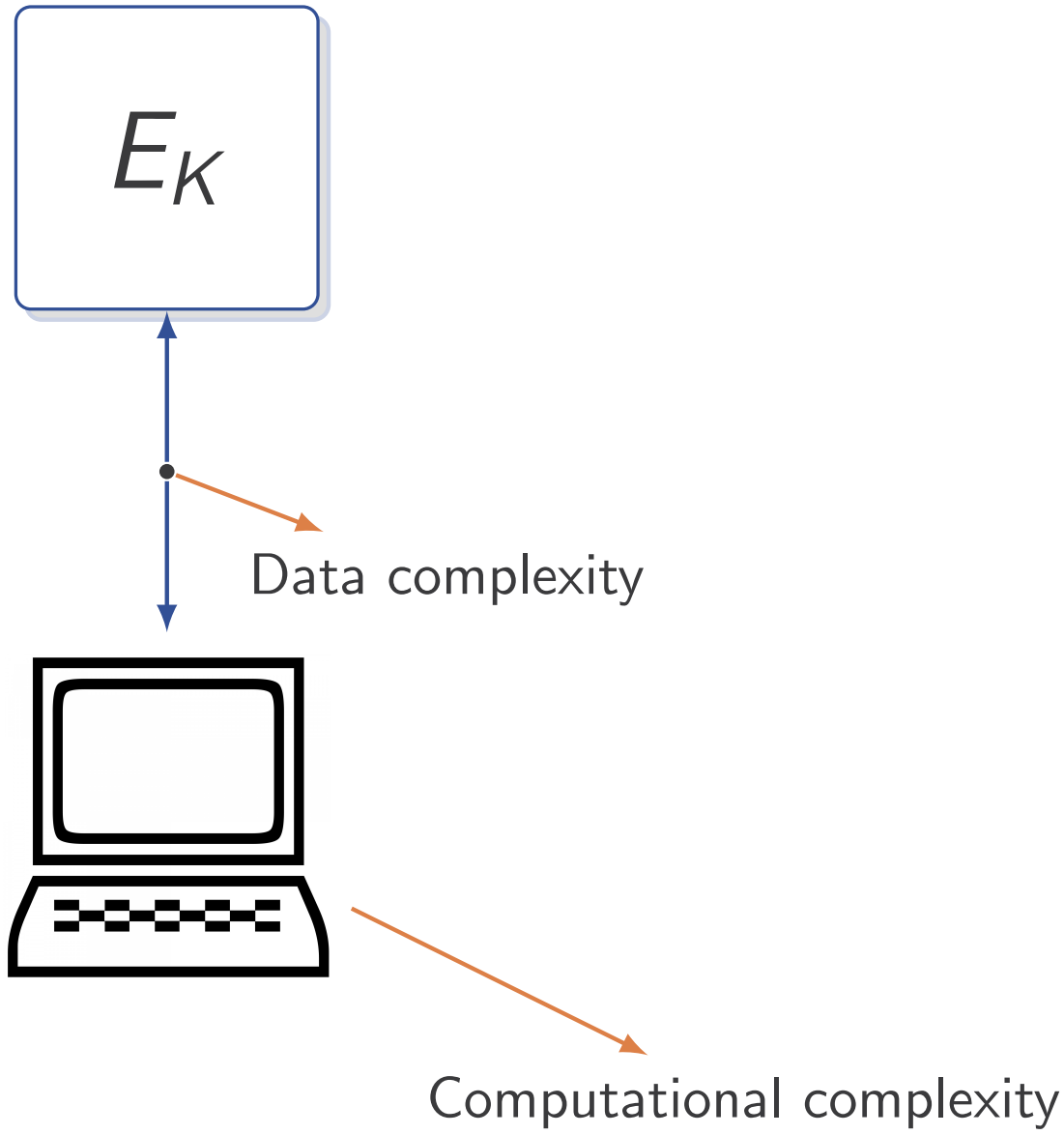


Example : Block Ciphers

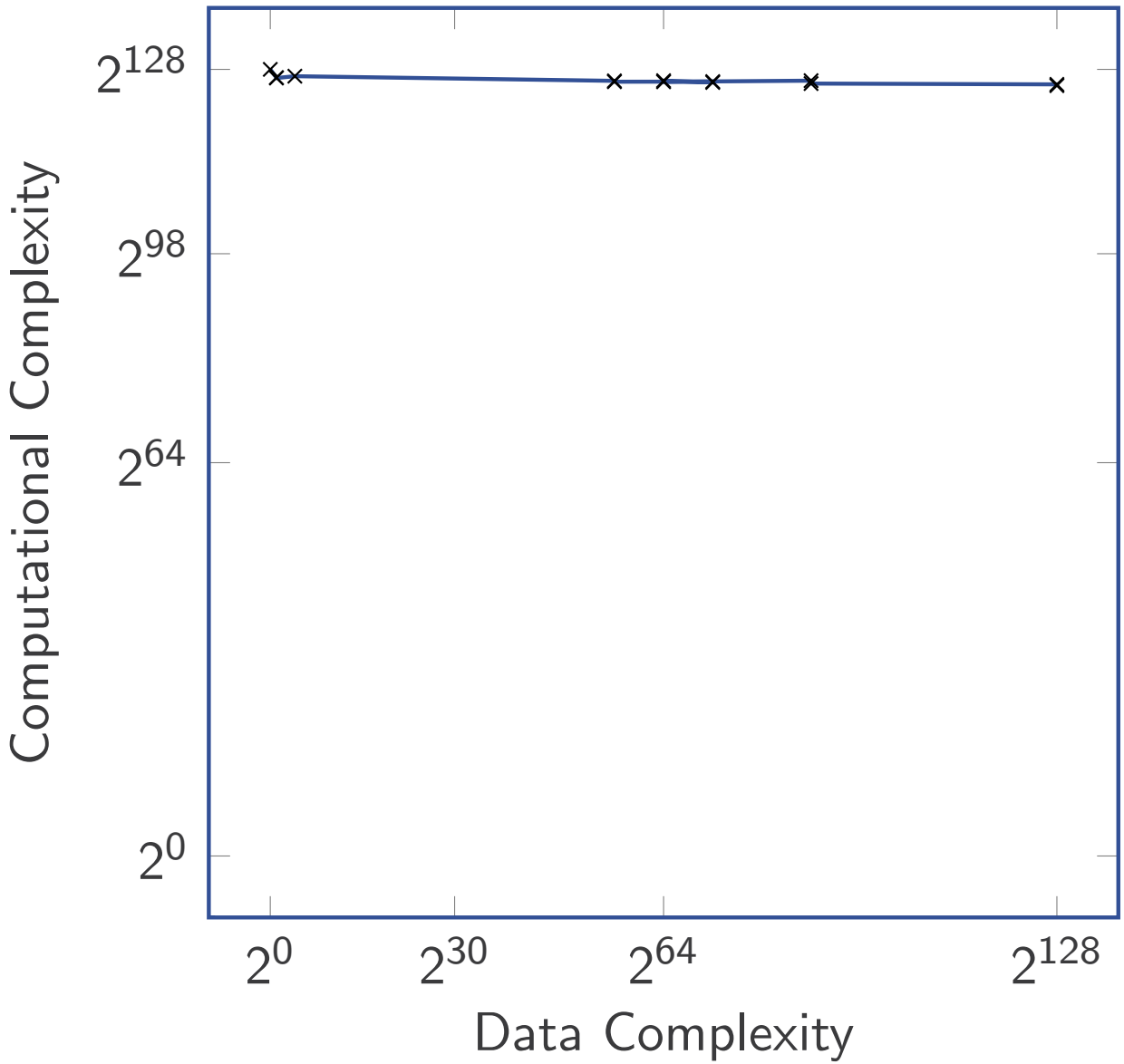


Computational complexity

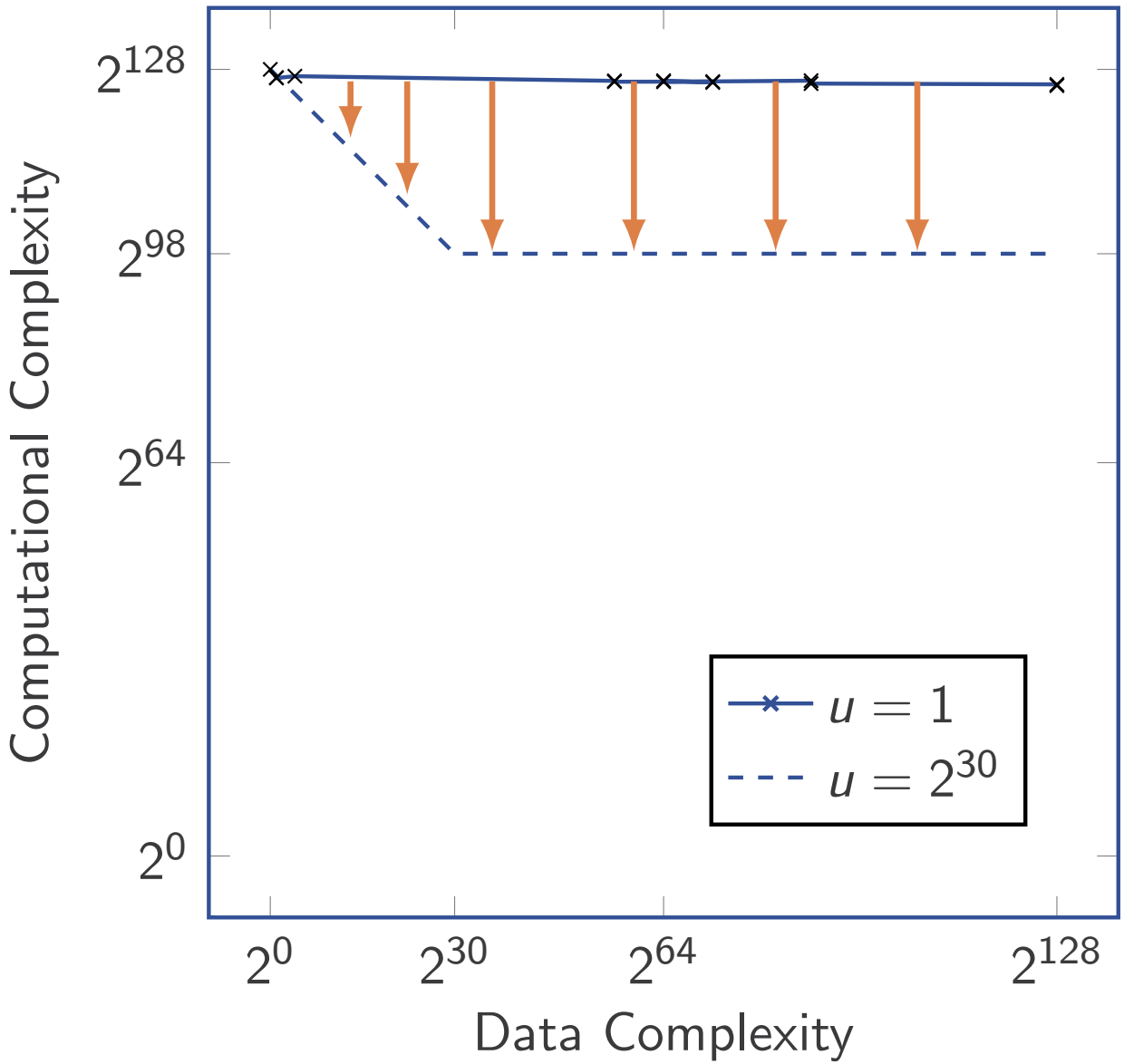
Example : Block Ciphers



Example: AES128 Key Recovery



Example: AES128 Key Recovery



Block Cipher Multi-key Attack

Precomputation:

$(L, E_L(P))$

Block Cipher Multi-key Attack

Precomputation:

$$(L, E_L(P))$$

Queries:

$$E_{K_1}(P), E_{K_2}(P), \ldots, E_{K_u}(P)\,.$$

Block Cipher Multi-key Attack

Precomputation:

$$(L, E_L(P))$$

Queries:

$$E_{K_1}(P), E_{K_2}(P), \ldots, E_{K_u}(P)\,.$$

Compare precomputed ciphertext with received ciphertext

Block Cipher Multi-key Attack

Precomputation:

$(L, E_L(P))$

Queries:

$E_{K_1}(P), E_{K_2}(P), \ldots, E_{K_u}(P)\,.$

Compare precomputed ciphertext with received ciphertext

Biham 2002 Information Processing Letters

Block Cipher Multi-key Attack

Precomputation:

$$(L, E_L(P))$$

Queries:

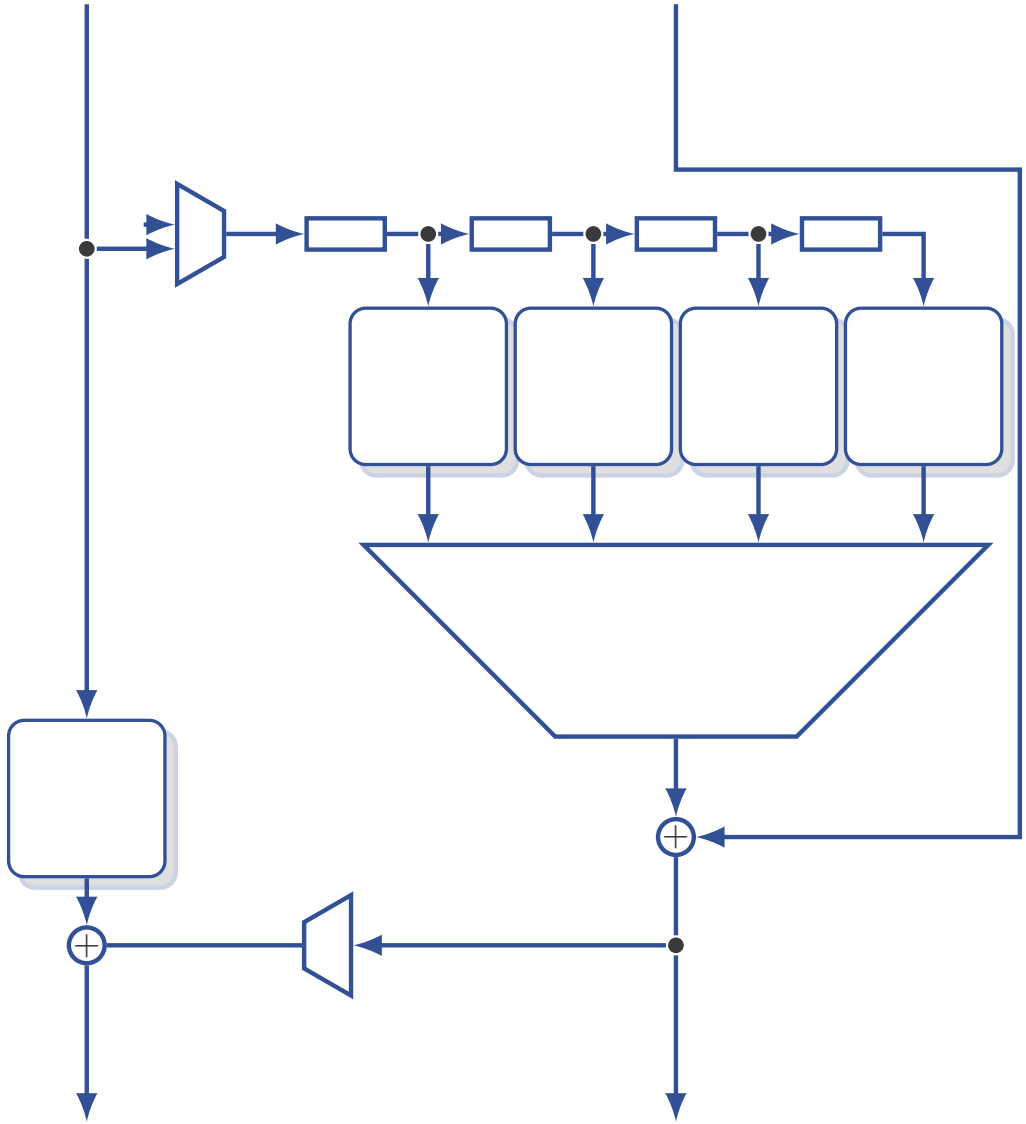
$$E_{K_1}(P), E_{K_2}(P), \ldots, E_{K_u}(P)\,.$$

Compare precomputed ciphertext with received ciphertext

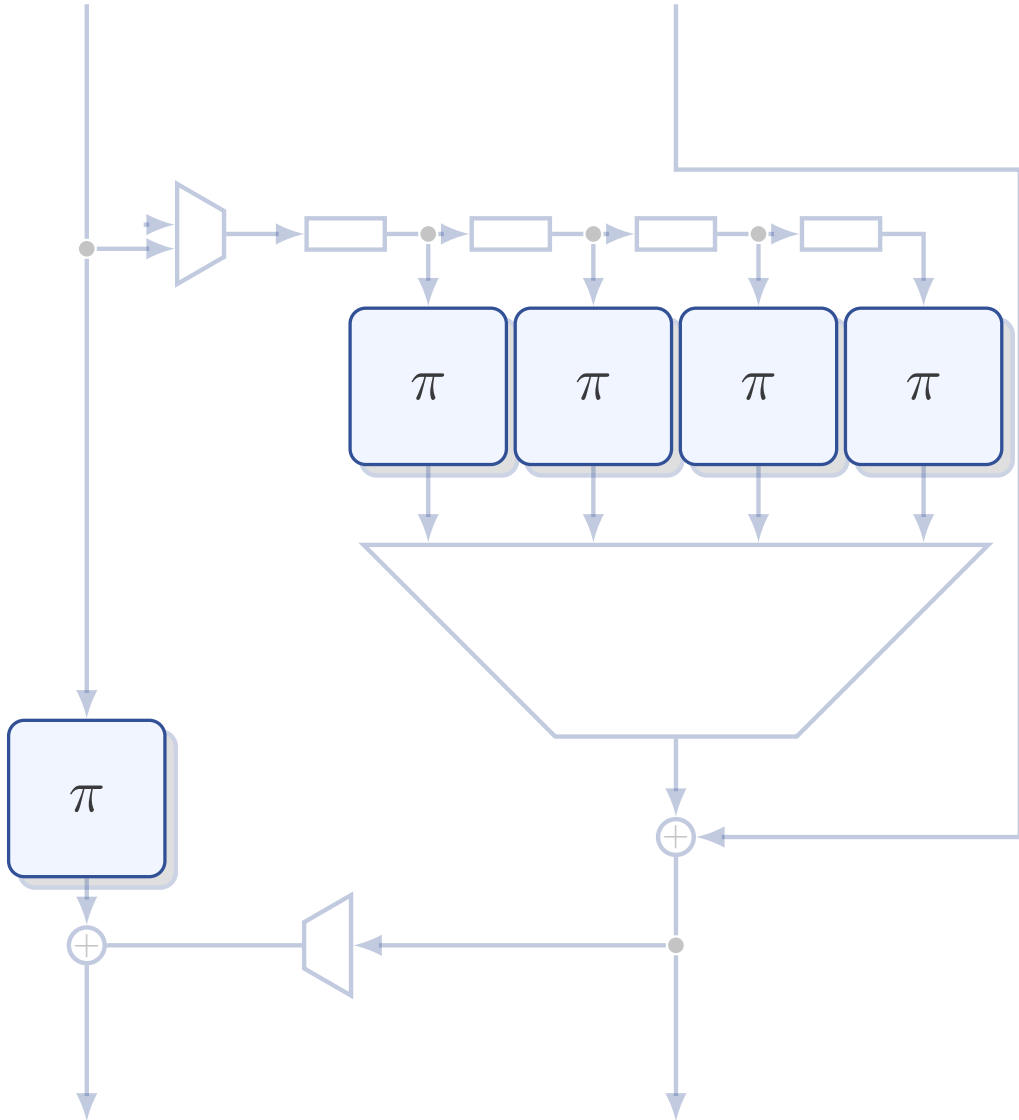
Biham 2002 Information Processing Letters

Biryukov, Mukhopadhyay, Sarkar, SAC 2005: time-memory-data trade-off

Example: GCM

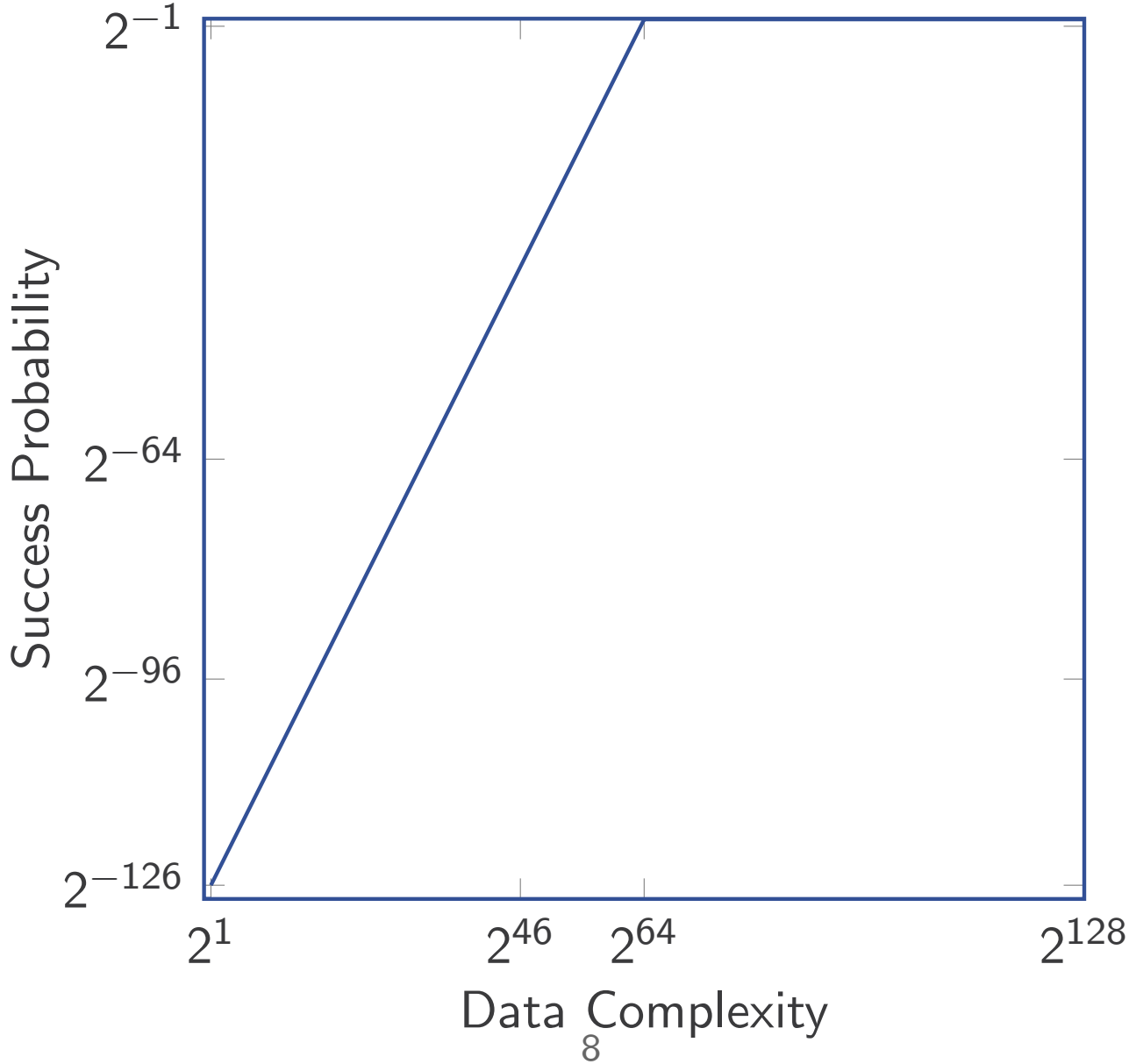


Example: GCM

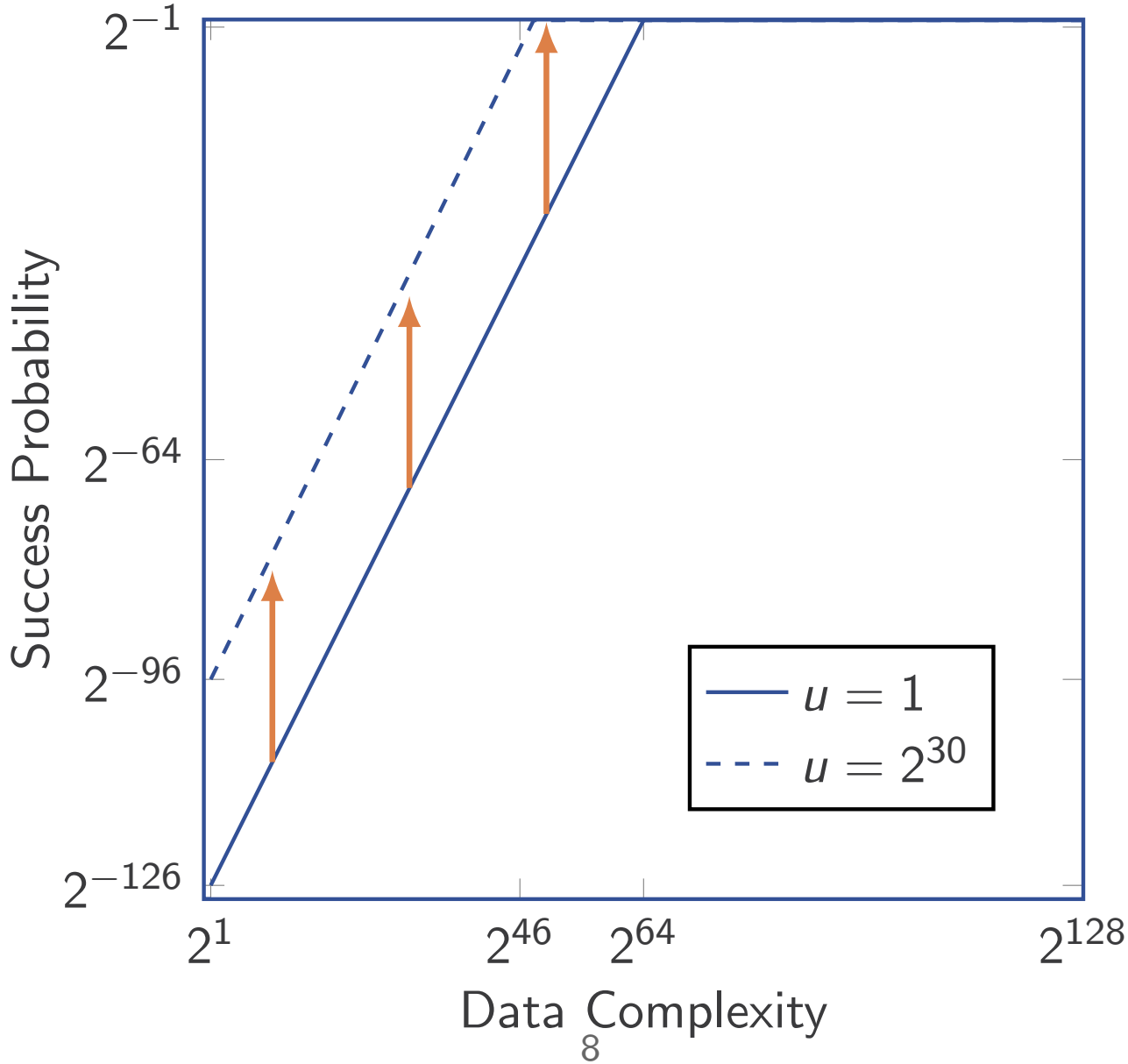


Biham attack inapplicable

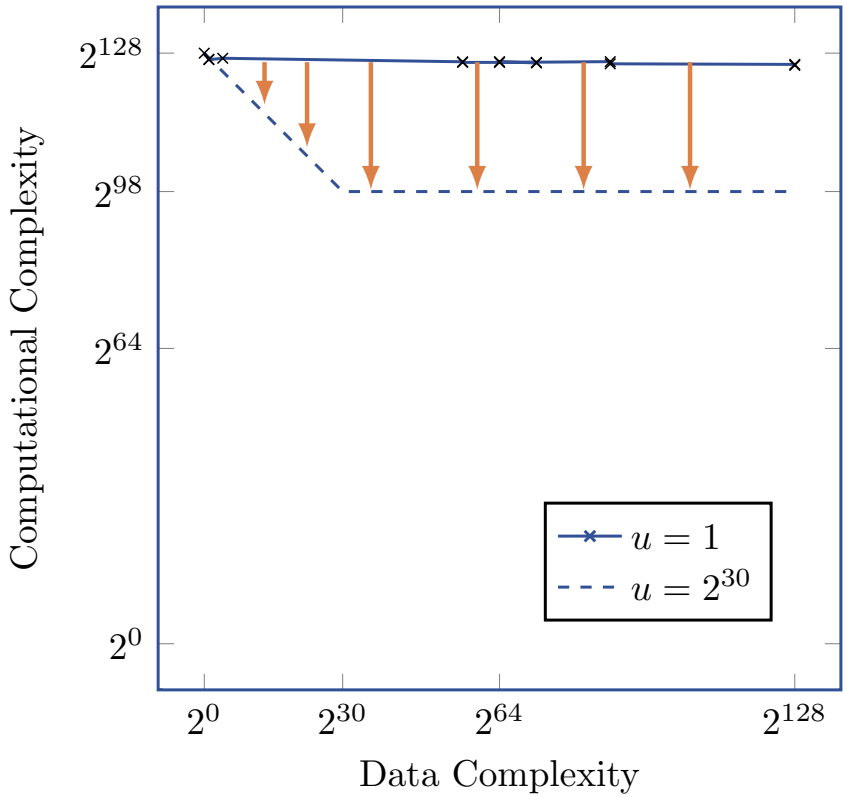
GCM Bound in the Multi-Key Setting

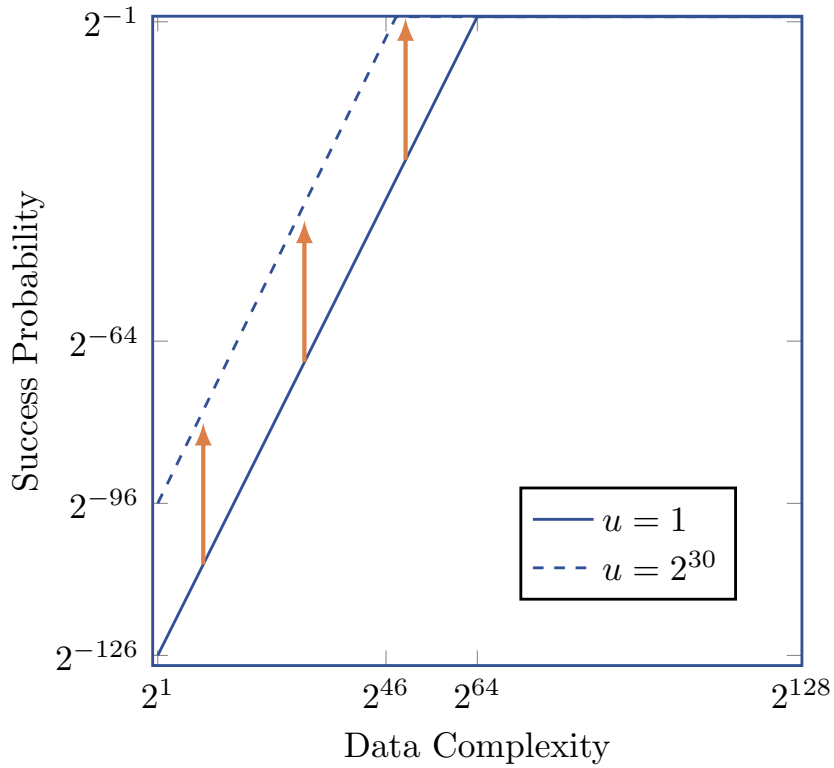


GCM Bound in the Multi-Key Setting

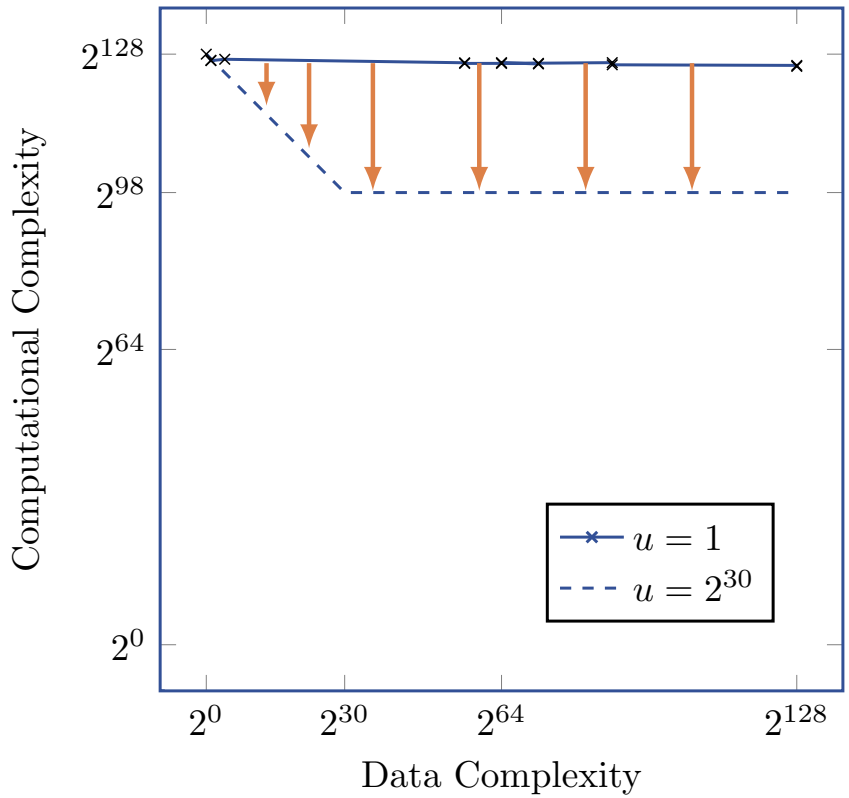


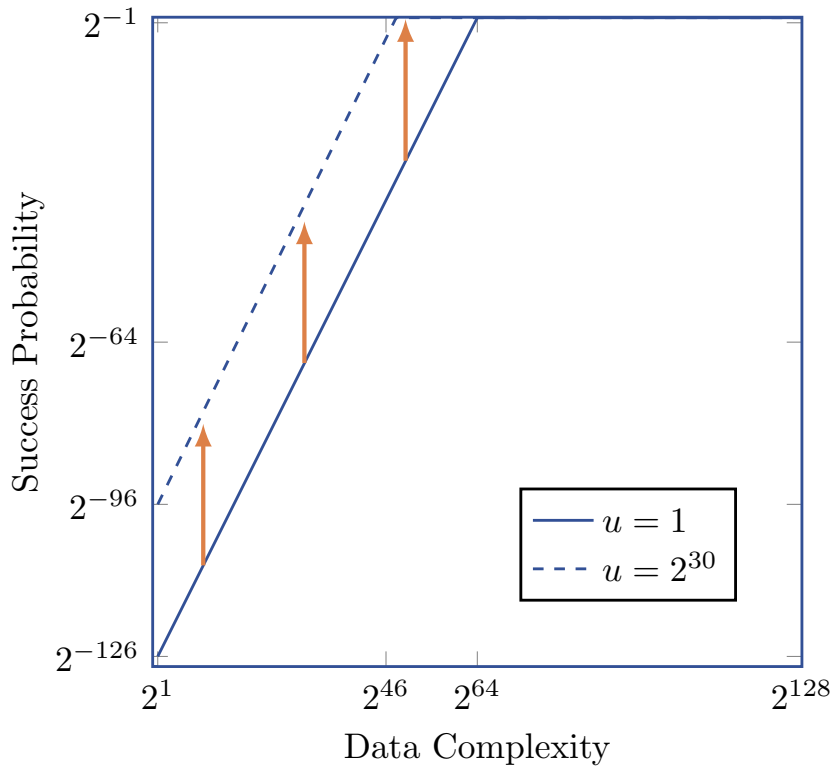
AES128 Multikey vs. GCM Multikey





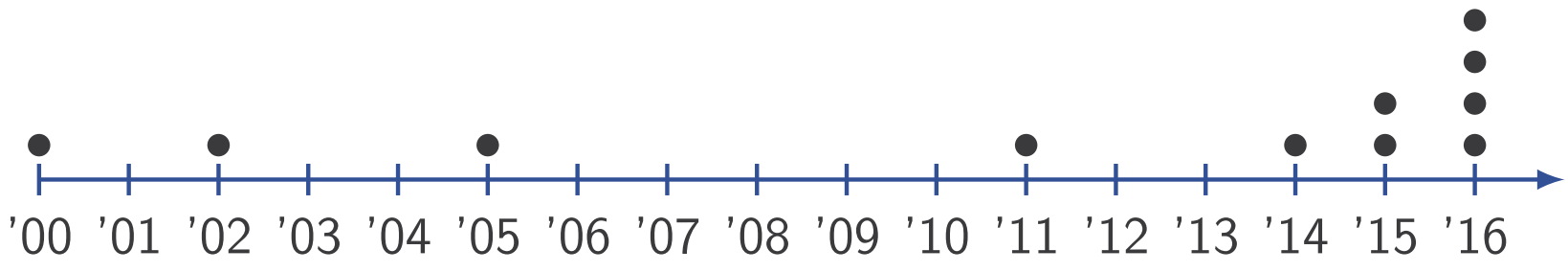
AES128 Multikey vs. GCM Multikey



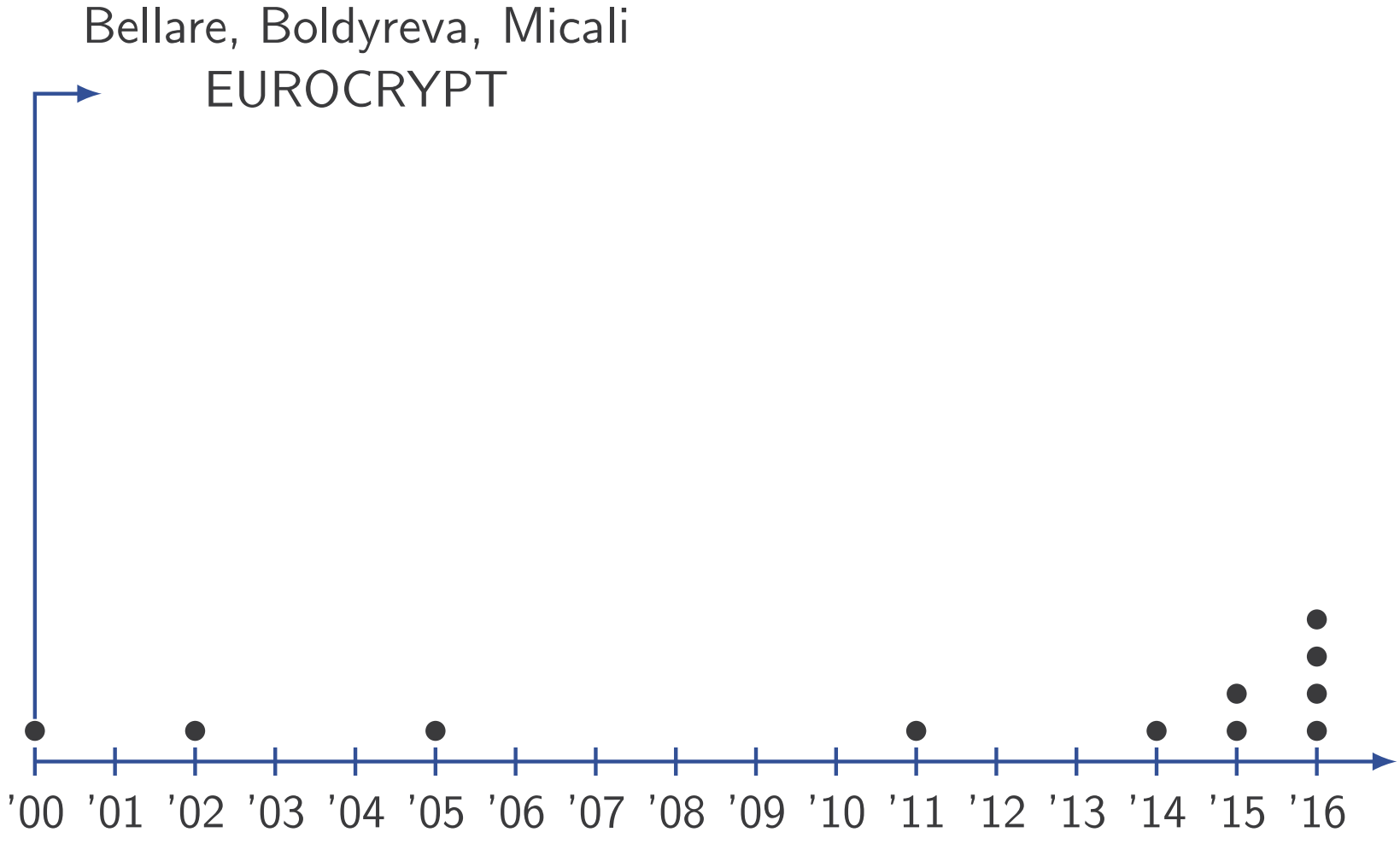


Matching attacks?

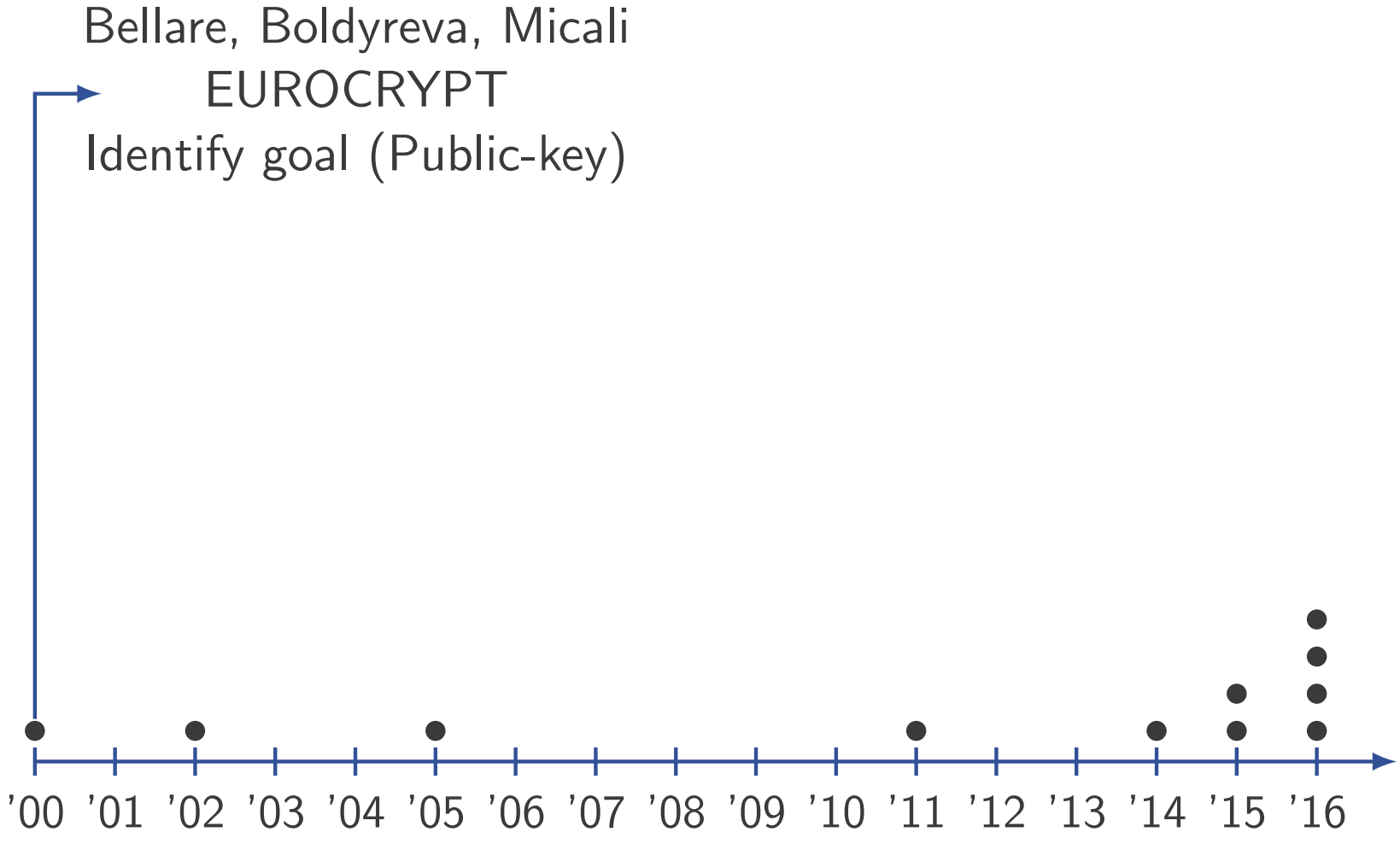
Work on Multi-Key Security in Symmetric Setting



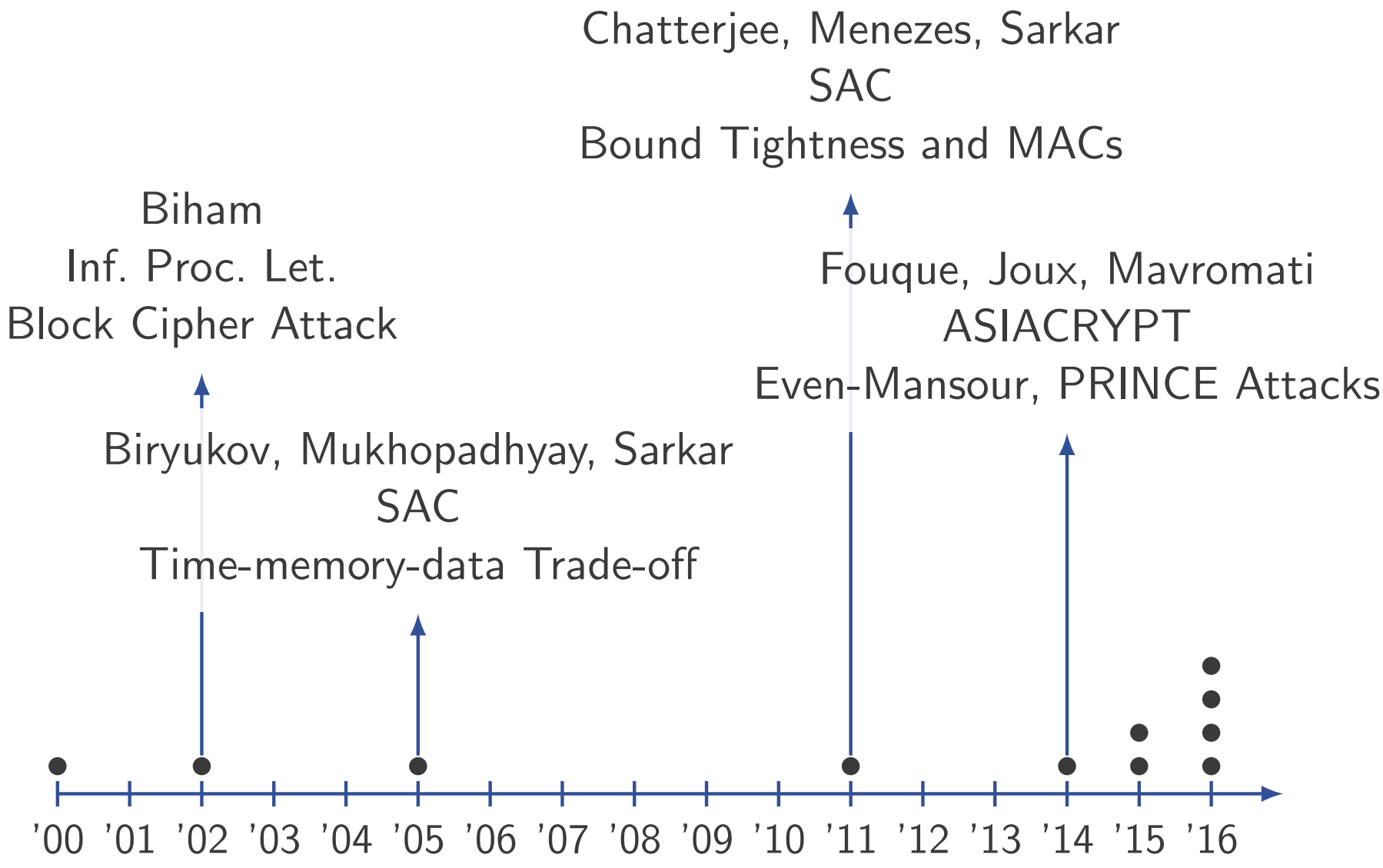
Work on Multi-Key Security in Symmetric Setting



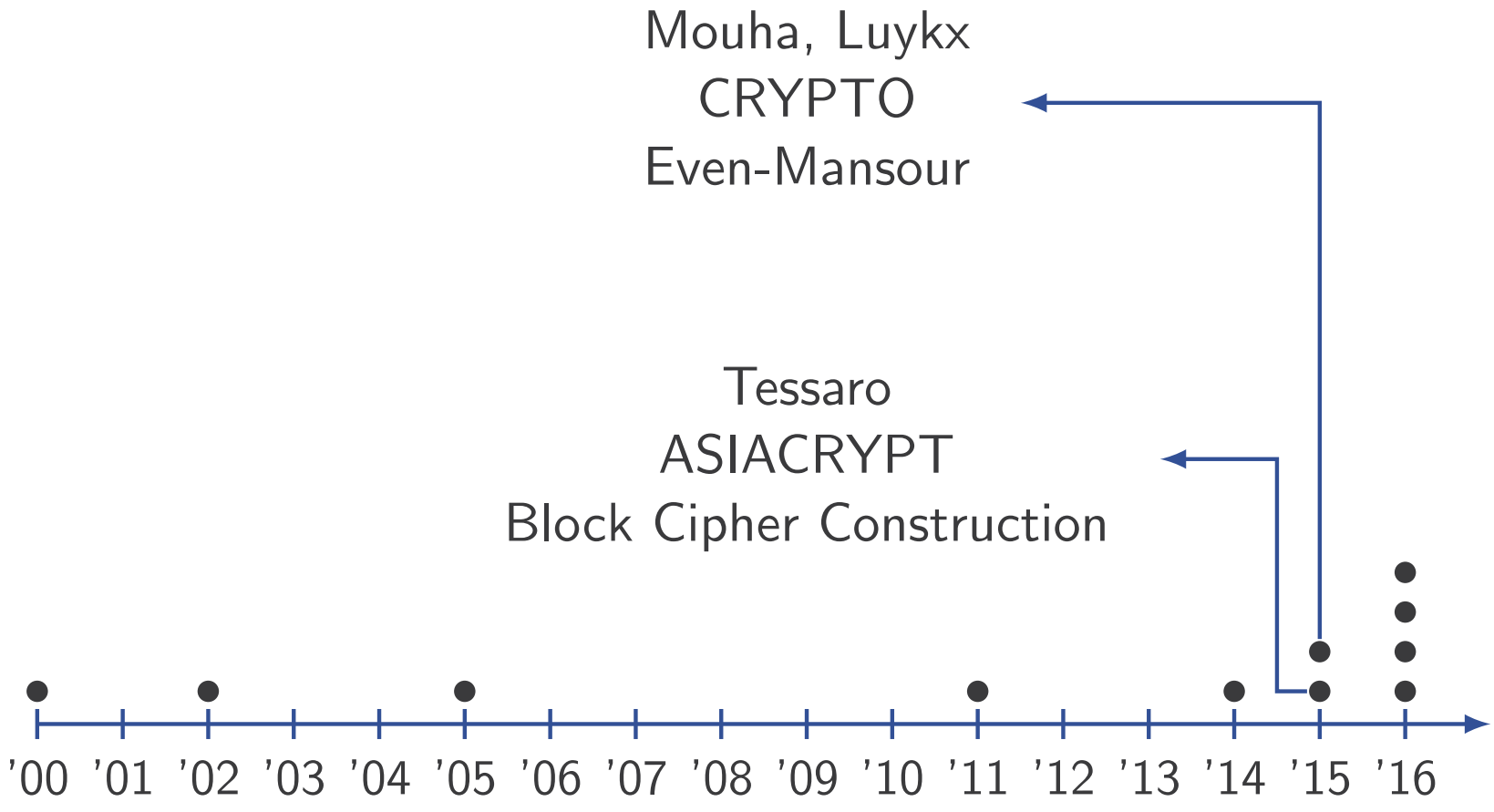
Work on Multi-Key Security in Symmetric Setting



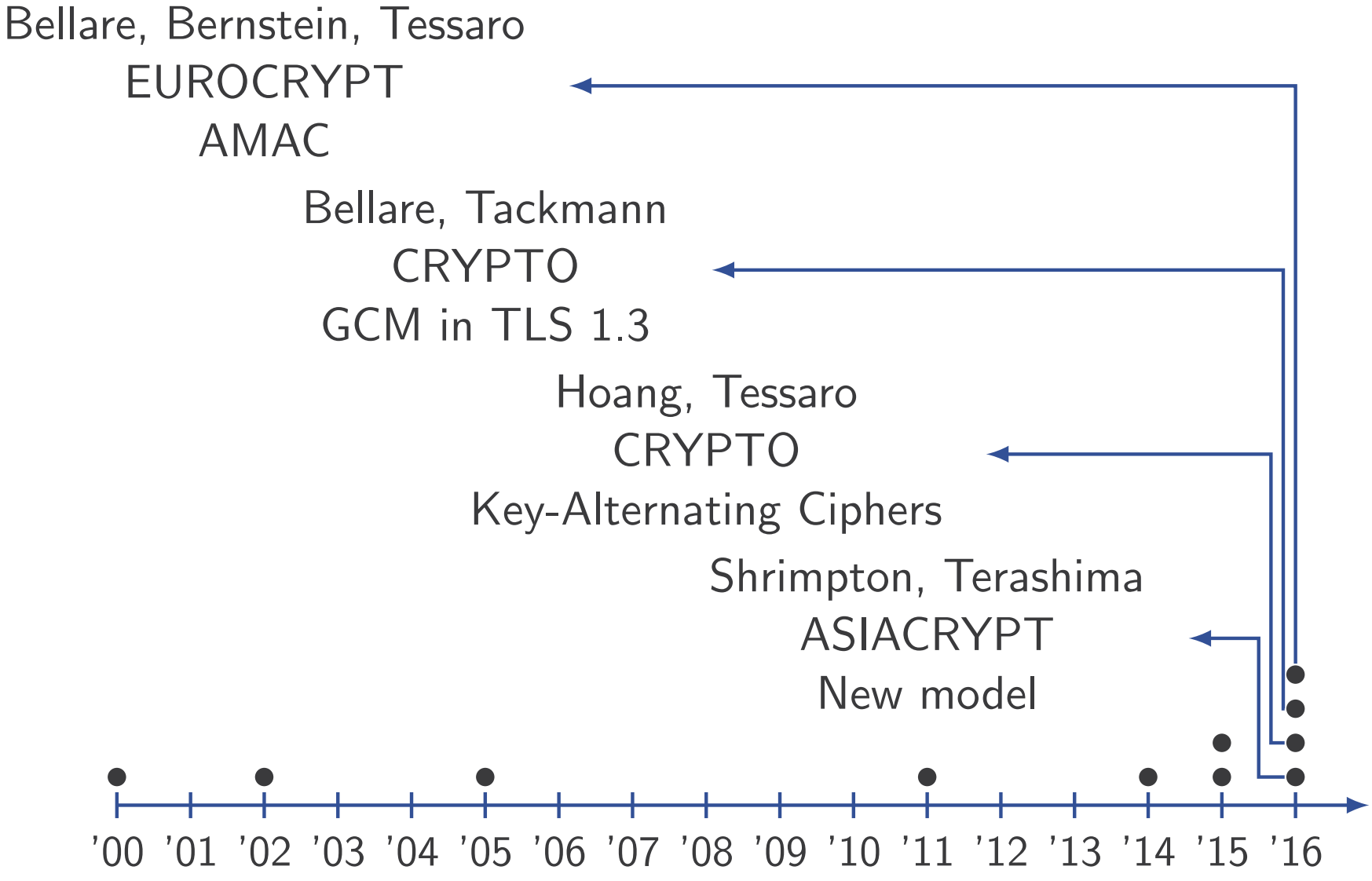
Work on Multi-Key Security in Symmetric Setting

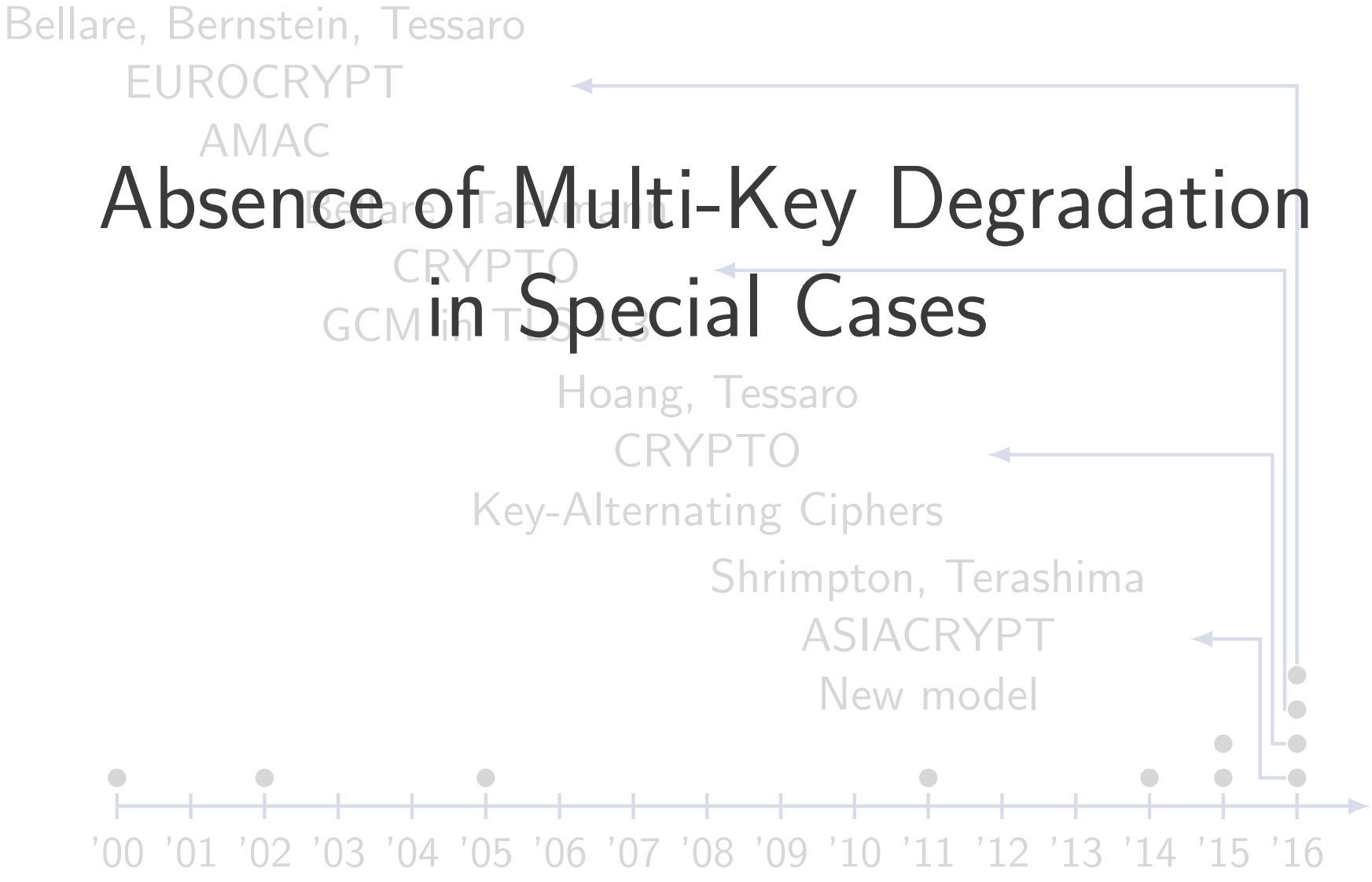


Work on Multi-Key Security in Symmetric Setting

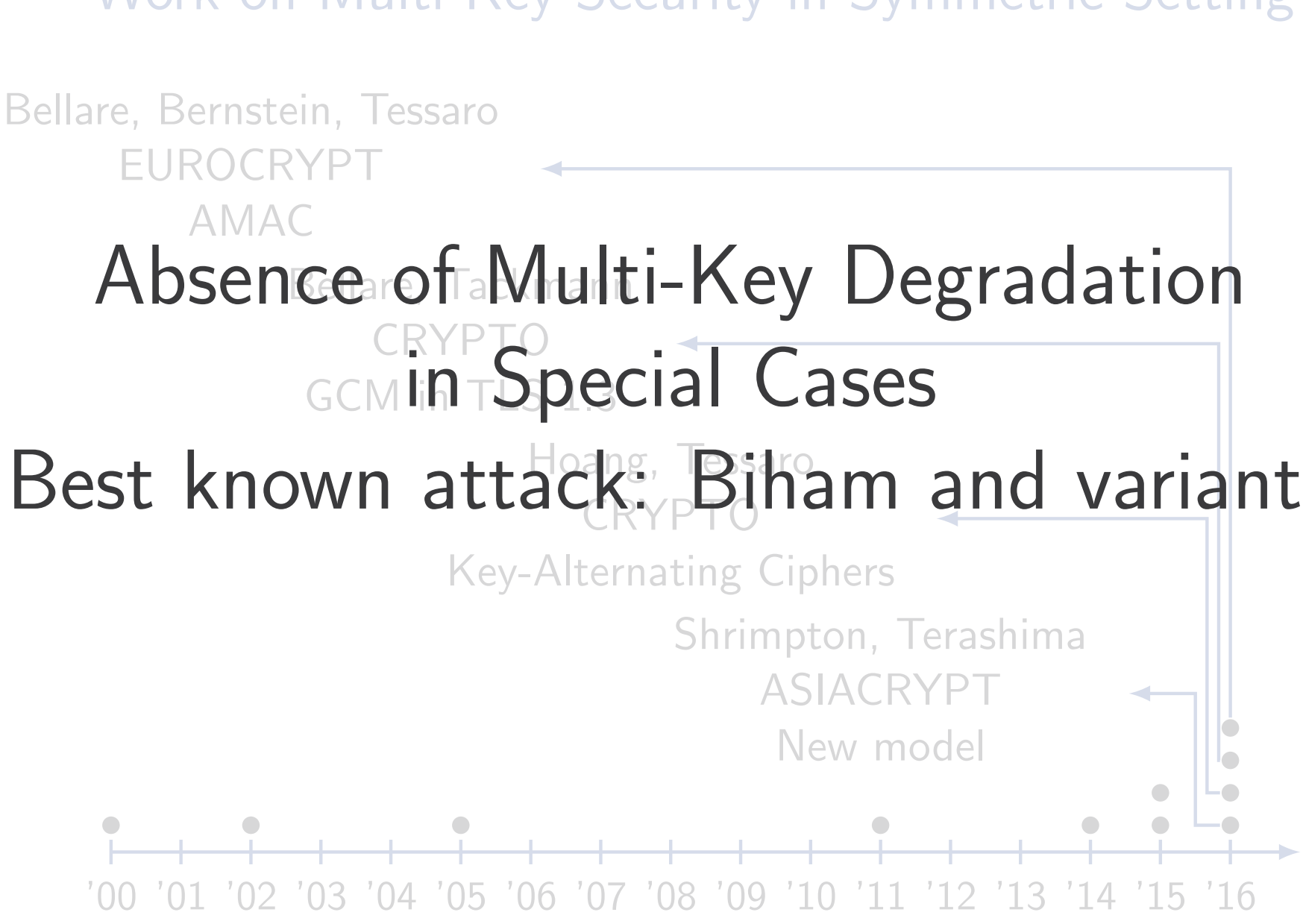


Work on Multi-Key Security in Symmetric Setting





Absence of Multi-Key Degradation in Special Cases

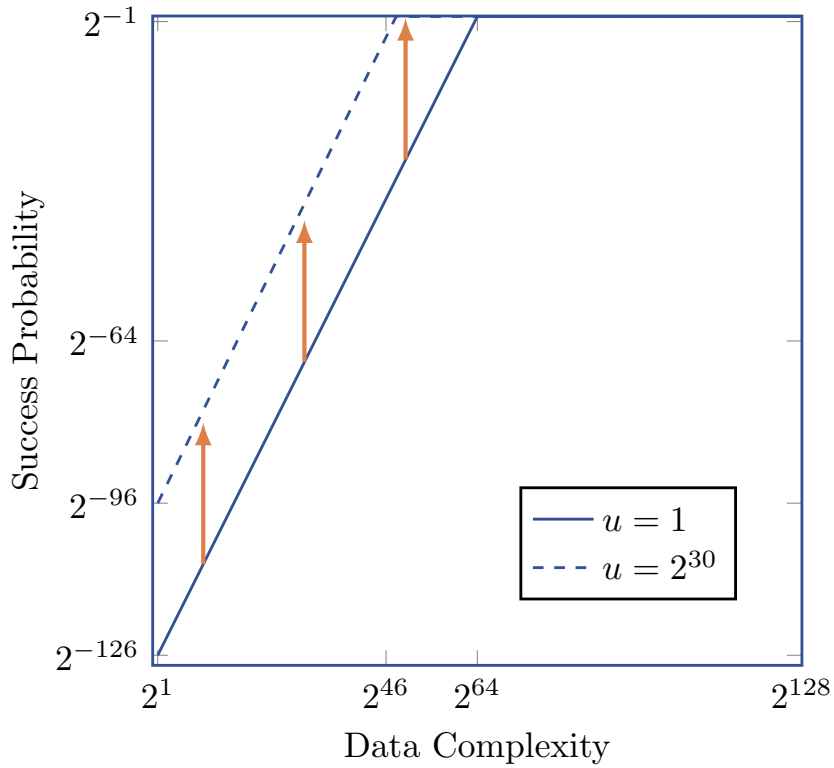


Absence of Multi-Key Degradation in Special Cases

Best known attack: Biham and variants

Our Work

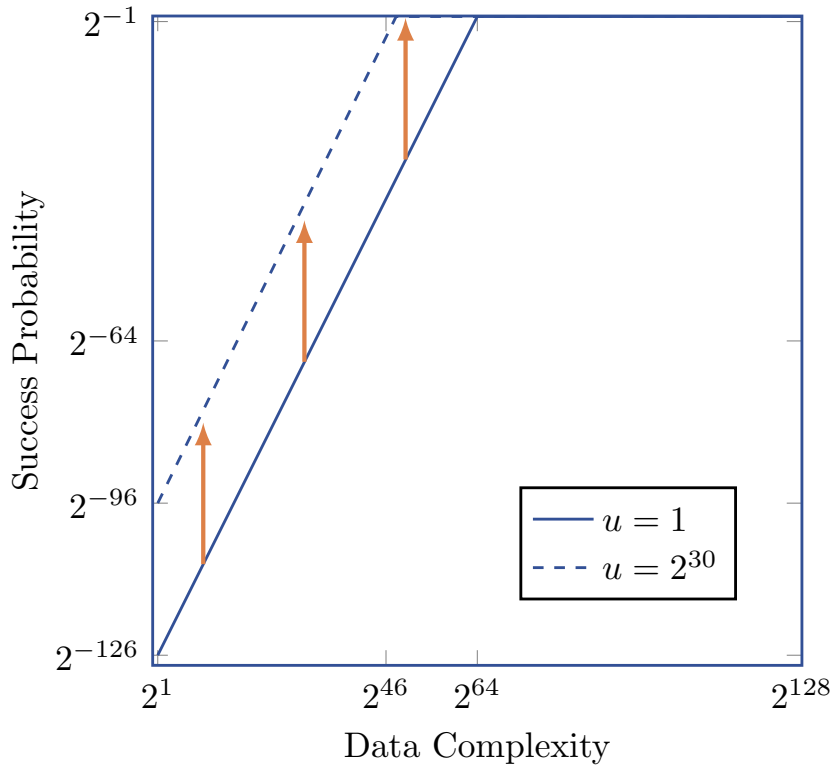
1. Set out to understand gap



Matching attacks?

Our Work

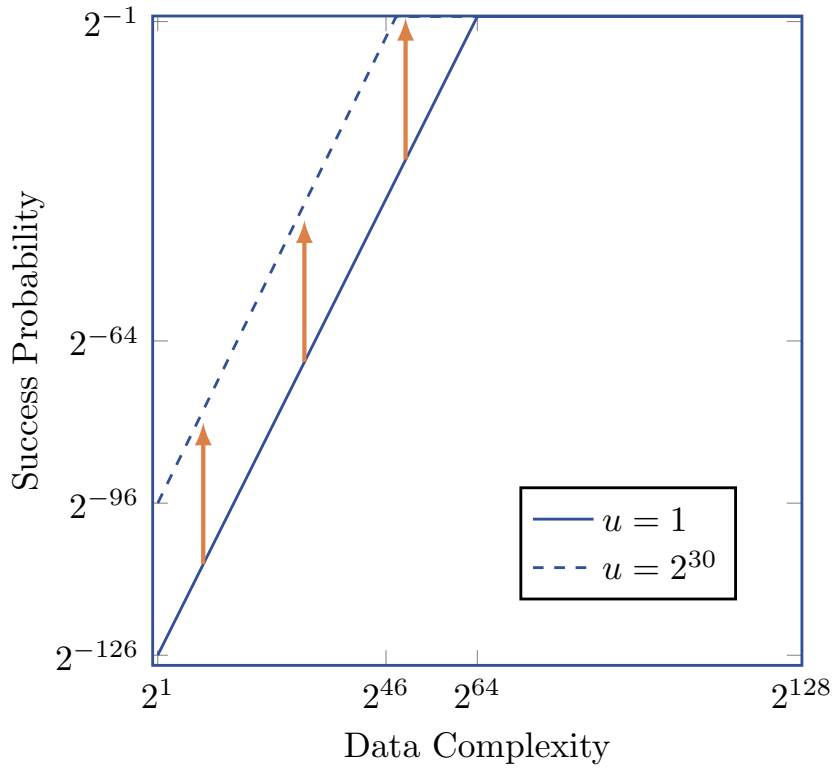
1. Set out to understand gap
2. Characterization of multi-key setting: necessary and sufficient condition for degradation



Matching attacks?

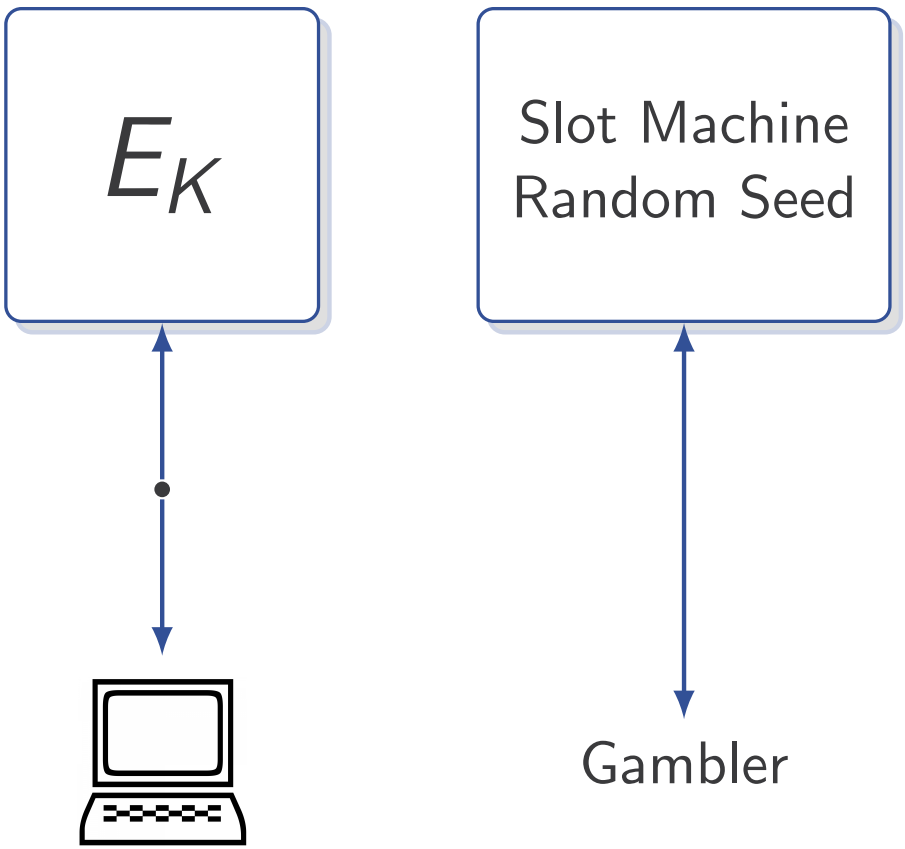
Our Work

1. Set out to understand gap
2. Characterization of multi-key setting: necessary and sufficient condition for degradation
3. Proved in abstract setting, applied to GCM

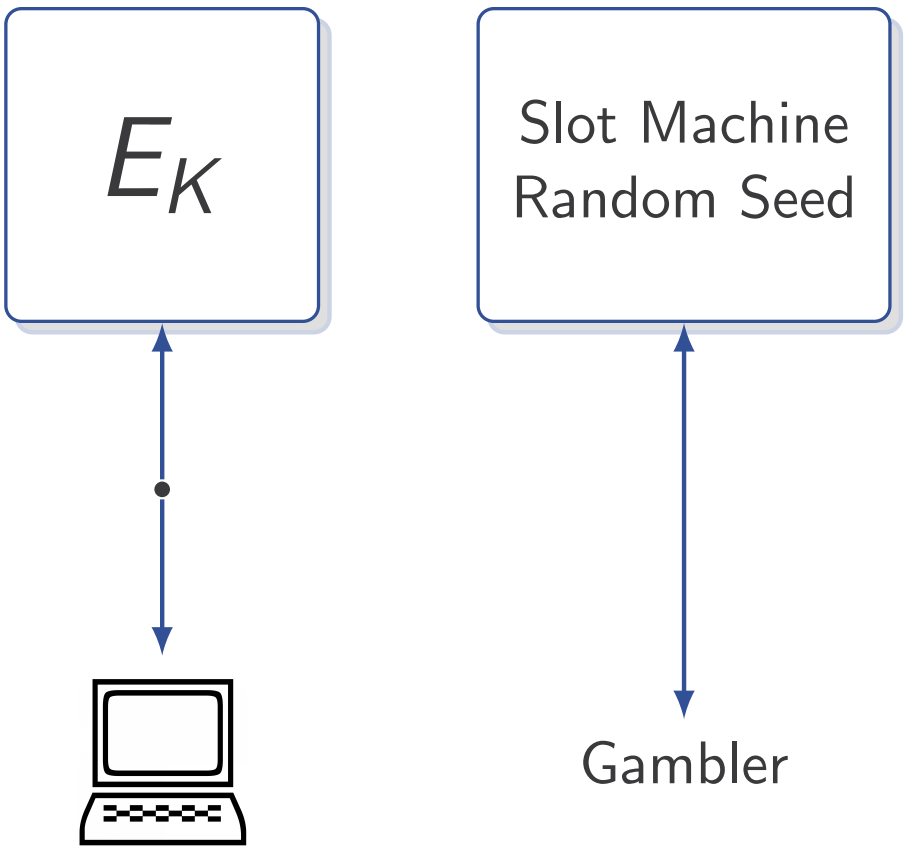


Matching attacks?

Slot Machine Scenario

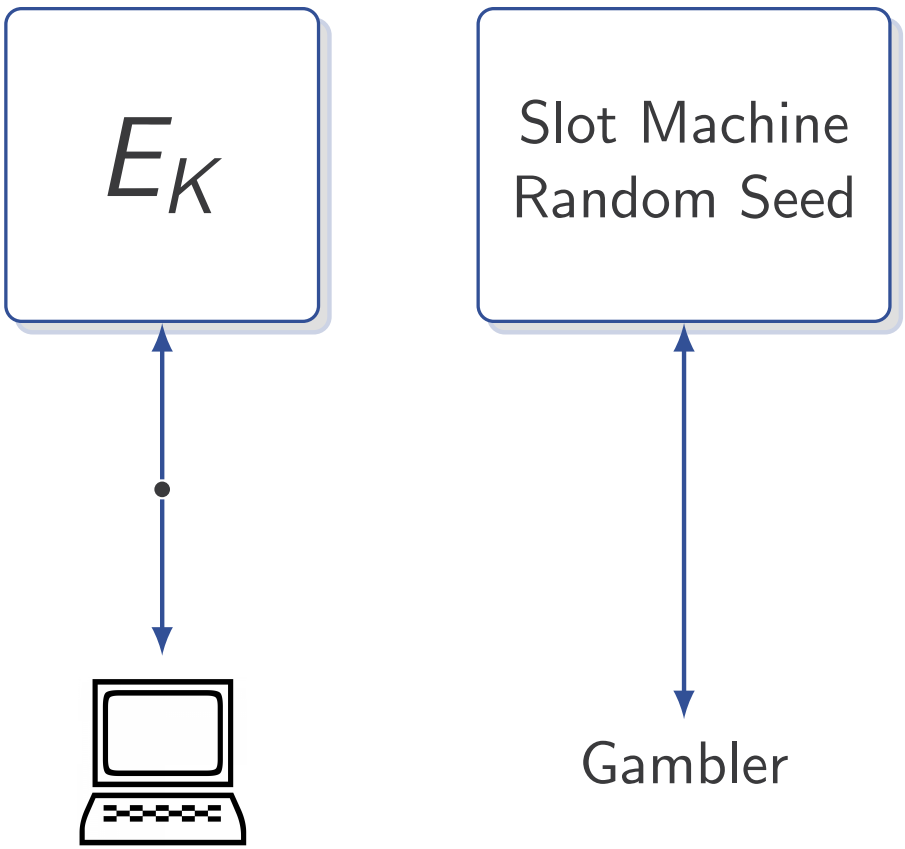


Slot Machine Scenario



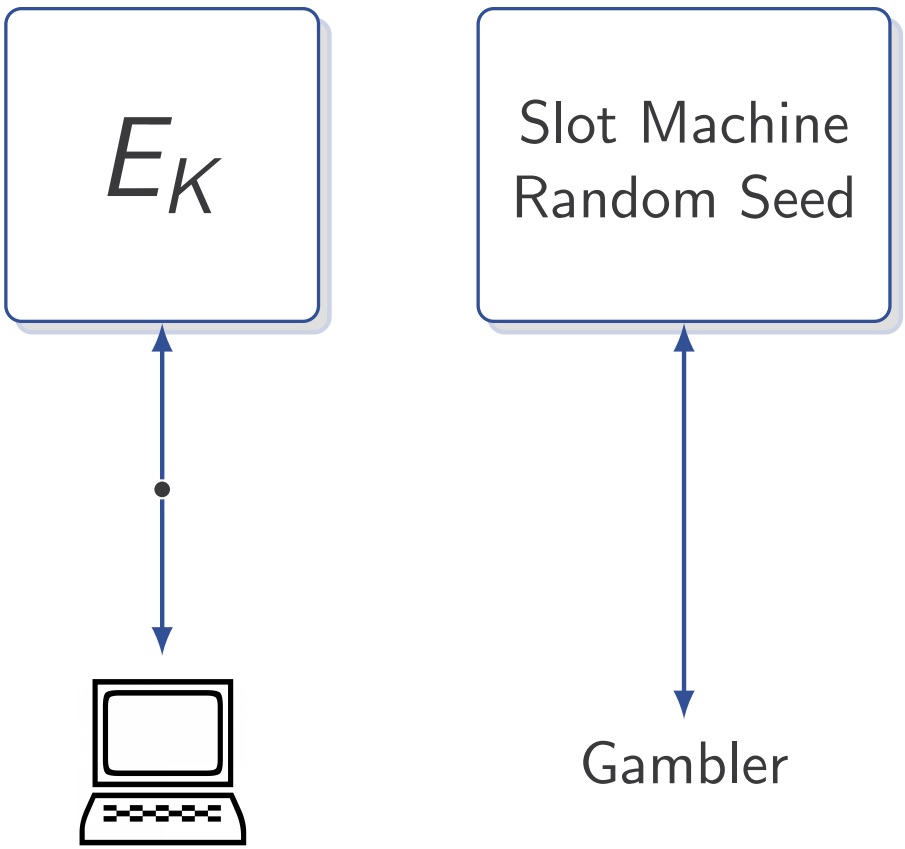
1. Data complexity = money

Slot Machine Scenario



1. Data complexity = money
2. Success = jackpot

Slot Machine Scenario



1. Data complexity = money
2. Success = jackpot
3. Single-key setting = access to one slot machine

Slot Machine Scenario

Casino Setting

500 coin budget

Casino Setting

500 coin budget, 100 slot machines

Casino Setting

500 coin budget, 100 slot machines
all 500 coins on 1 machine
vs
500 coins distributed somehow over 100 machines

Casino Setting

500 coin budget, 100 slot machines
all 500 coins on 1 machine
vs
500 coins distributed somehow over 100 machines

"spending 500 coins on 100 slot machines gives you factor 100 higher success than spending 500 coins on one slot machine"

Casino Setting

500 coin budget, 100 slot machines
all 500 coins on 1 machine
vs
500 coins distributed somehow over 100 machines

"spending 500 coins on 100 slot machines gives you factor 100 higher success than spending 500 coins on one slot machine"

Counter-intuitive, ever possible?

Casino Setting

500 coin budget, 100 slot machines
all 500 coins on 1 machine
vs
500 coins distributed somehow over 100 machines

"spending 500 coins on 100 slot machines gives you factor 100 higher success than spending 500 coins on one slot machine"

Counter-intuitive, ever possible? Yes.

Casino Setting

500 coin budget, 100 slot machines
all 500 coins on 1 machine
vs
500 coins distributed somehow over 100 machines

"spending 500 coins on 100 slot machines gives you factor 100 higher success than spending 500 coins on one slot machine"

Counter-intuitive, ever possible? Yes.

1. Assume slot machine is "lucky" with some probability

Casino Setting

500 coin budget, 100 slot machines
all 500 coins on 1 machine
vs
500 coins distributed somehow over 100 machines

"spending 500 coins on 100 slot machines gives you factor 100 higher success than spending 500 coins on one slot machine"

Counter-intuitive, ever possible? Yes.

1. Assume slot machine is "lucky" with some probability
2. One slot machine: either lucky or not.

Casino Setting

500 coin budget, 100 slot machines
all 500 coins on 1 machine
vs
500 coins distributed somehow over 100 machines

"spending 500 coins on 100 slot machines gives you factor 100 higher success than spending 500 coins on one slot machine"

Counter-intuitive, ever possible? Yes.

1. Assume slot machine is "lucky" with some probability
2. One slot machine: either lucky or not.
3. One hundred slot machines: find lucky machine, focus on that one

Lucky Machines in Cryptography

Weak keys

Lucky Machines in Cryptography

Weak keys

Midori64: 2^{32} weak keys out of 2^{128}, identifiable with one query (Guo et al. 2015)

Lucky Machines in Cryptography

Weak keys

Midori64: 2^{32} weak keys out of 2^{128}, identifiable with one query (Guo et al. 2015)

Weak-key recovery: computational complexity 2^{16}, data complexity 2.

Lucky Machines in Cryptography

Weak keys

Midori64: 2^{32} weak keys out of 2^{128}, identifiable with one query (Guo et al. 2015)

Weak-key recovery: computational complexity 2^{16}, data complexity 2.

Table: Midori64 key recovery

	$u = 1$	$u = 2^{16}$
Computational cost	2^{16}	2^{17}
Data cost	2	$u + 2$
Success Estimate	2^{-96}	2^{-80}

Understanding Multi-Key Security

1. Biham key-recovery attack if key size is not too big

Understanding Multi-Key Security

1. Biham key-recovery attack if key size is not too big
2. Exploit Weak instances if present

Understanding Multi-Key Security

1. Biham key-recovery attack if key size is not too big
2. Exploit Weak instances if present
3. What else can happen?

Necessary and Sufficient Condition

Setting 1
One slot machine
300 coins

Necessary and Sufficient Condition

Setting 1
One slot machine
300 coins

Setting 2
One slot machine
300 coins

Necessary and Sufficient Condition

Setting 1
One slot machine
300 coins

Setting 2
One slot machine
300 coins
Friend has played on slot machine with 200 coins
Gives you history

Necessary and Sufficient Condition

Setting 1
One slot machine
300 coins

Setting 2
One slot machine
300 coins
Friend has played on slot machine with 200 coins
Gives you history

If

Necessary and Sufficient Condition

Setting 1
One slot machine
300 coins

Setting 2
One slot machine
300 coins
Friend has played on slot machine with 200 coins
Gives you history

If

$$\text{Setting 1 Jackpot probability} \leq \text{Setting 2 Jackpot probability}$$

Necessary and Sufficient Condition

Setting 1
One slot machine
300 coins

Setting 2
One slot machine
300 coins
Friend has played on slot machine with 200 coins
Gives you history

If

Setting 1 Jackpot probability $\leq$ Setting 2 Jackpot probability

for all histories below some cost

Necessary and Sufficient Condition

Setting 1
One slot machine
300 coins

Setting 2
One slot machine
300 coins
Friend has played on slot machine with 200 coins
Gives you history

If

Setting 1 Jackpot probability $\leq$ Setting 2 Jackpot probability

for all histories below some cost

then no advantage interacting with multiple slot machines

Lucky Machines Excluded

Setting 1
One slot machine

Setting 2
One slot machine

Lucky Machines Excluded

Setting 1
One slot machine

Setting 2
One slot machine
Friend tells you the machine is *not* lucky

Lucky Machines Excluded

Setting 1
One slot machine

Setting 2
One slot machine
Friend tells you the machine is *not* lucky

Setting 1: possibility of interacting with lucky machine
$\Rightarrow$ jackpot probability might be higher

Translation to Oracles and Games

Setting 1
An oracle and game
maximum q queries

Setting 2
An oracle and game
maximum q queries
A transcript representing past history

Translation to Oracles and Games

Setting 1
An oracle and game
maximum q queries

Setting 2
An oracle and game
maximum q queries
A transcript representing past history

If

Setting 1 Adversarial Sucess $\leq$
Setting 2 Adversarial Success given transcript is satisfied

for all transcripts below some cost

then no advantage interacting with multiple oracle instances

Proof Intuition

1. Optimal adversaries gain no advantage in having one oracle's inputs depend on another oracle's outputs

Proof Intuition

1. Optimal adversaries gain no advantage in having one oracle's inputs depend on another oracle's outputs
2. Adversary interacts with two oracles, makes a query to one of them

Proof Intuition

1. Optimal adversaries gain no advantage in having one oracle's inputs depend on another oracle's outputs
2. Adversary interacts with two oracles, makes a query to one of them
3. It has information on one of them $\Rightarrow$ better to stick to that oracle

Proof Intuition

1. Optimal adversaries gain no advantage in having one oracle's inputs depend on another oracle's outputs
2. Adversary interacts with two oracles, makes a query to one of them
3. It has information on one of them $\Rightarrow$ better to stick to that oracle

Converse

1. There is a "bad" transcript for which it is better to start over

Proof Intuition

1. Optimal adversaries gain no advantage in having one oracle's inputs depend on another oracle's outputs
2. Adversary interacts with two oracles, makes a query to one of them
3. It has information on one of them $\Rightarrow$ better to stick to that oracle

Converse

1. There is a "bad" transcript for which it is better to start over
2. Attack one oracle, if bad transcript, switch to another oracle.

Proof Intuition

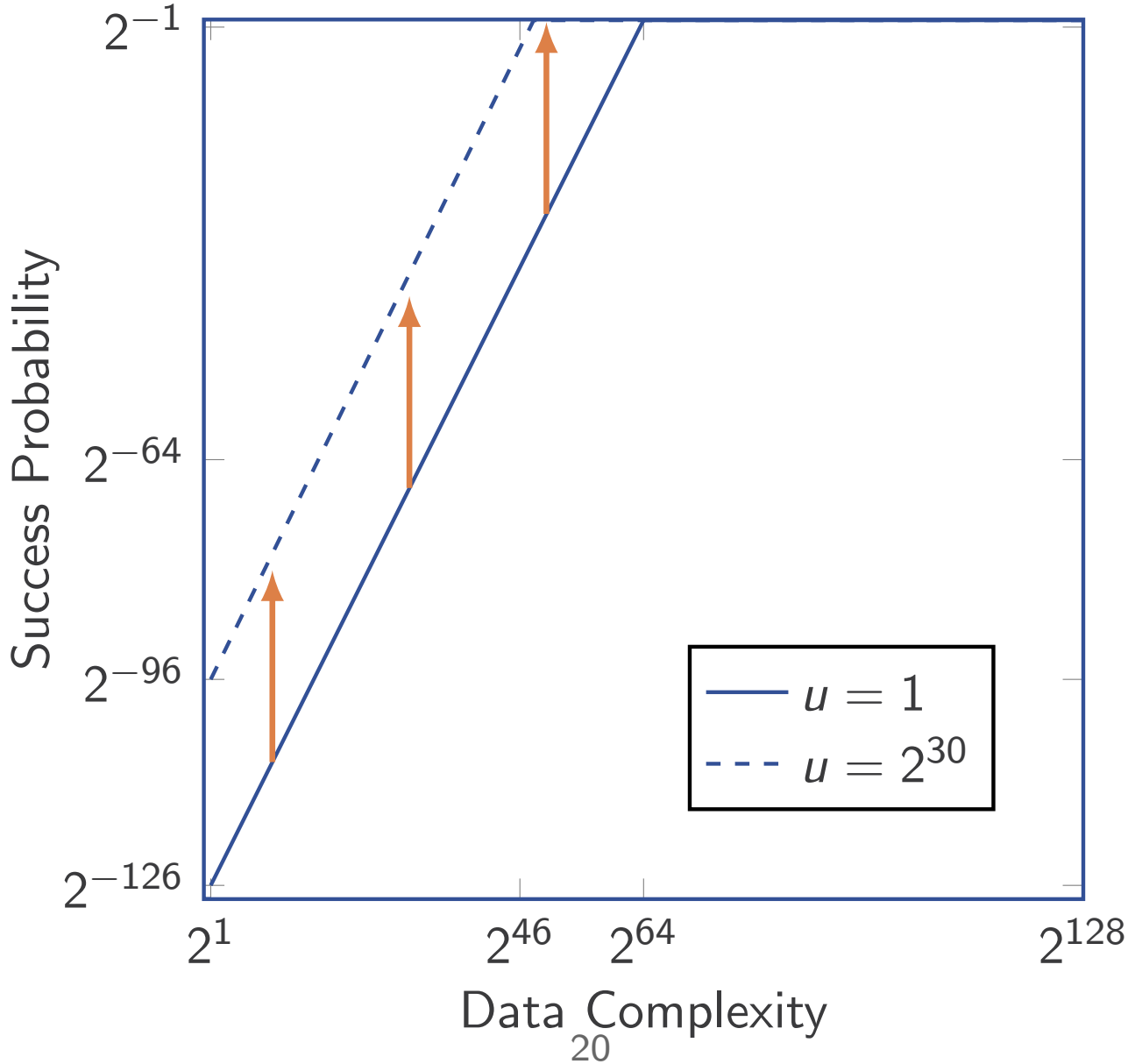
1. Optimal adversaries gain no advantage in having one oracle's inputs depend on another oracle's outputs
2. Adversary interacts with two oracles, makes a query to one of them
3. It has information on one of them $\Rightarrow$ better to stick to that oracle

Converse

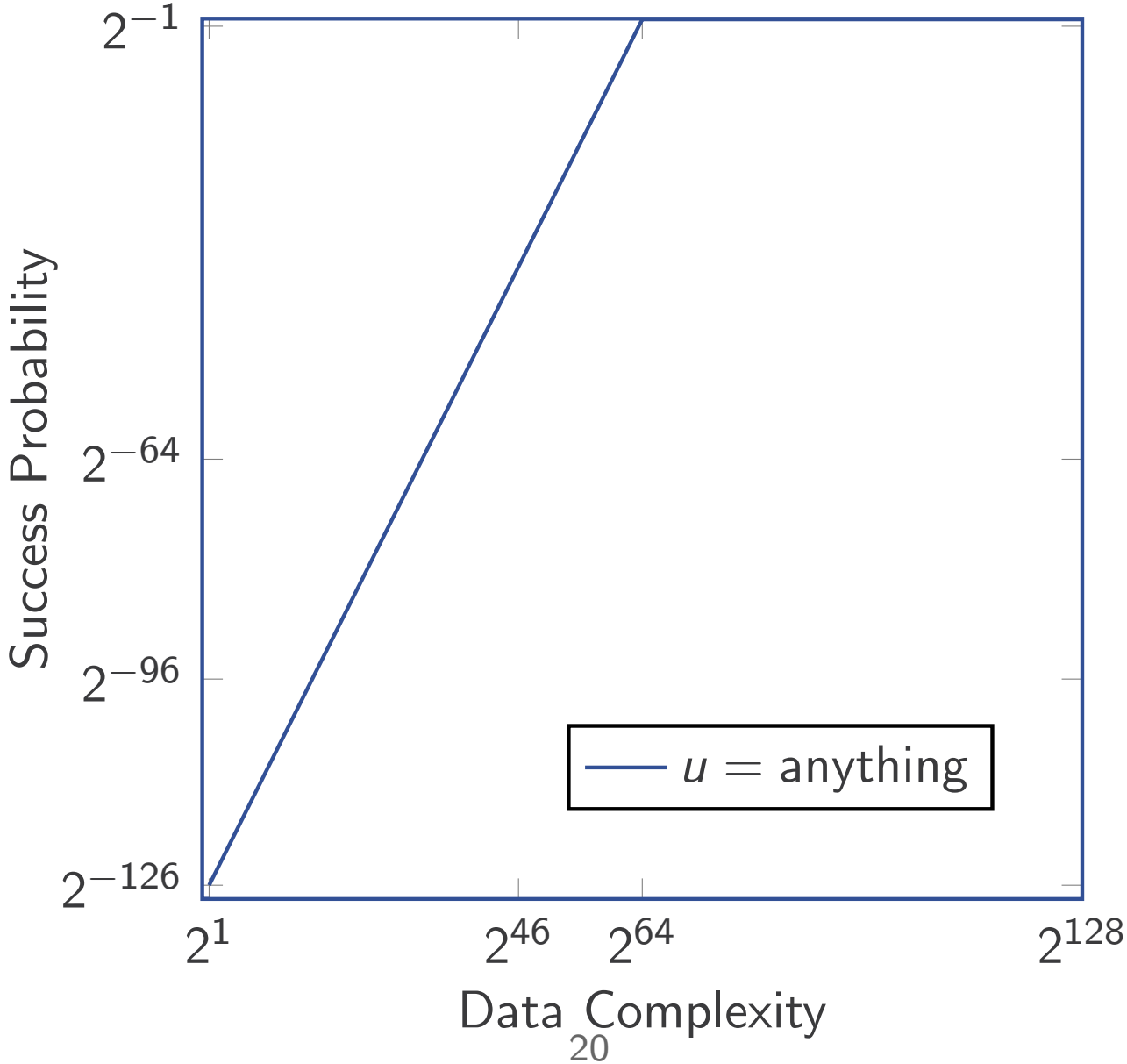
1. There is a "bad" transcript for which it is better to start over
2. Attack one oracle, if bad transcript, switch to another oracle.

This is only intuition, proper formalization introduces subtleties!
(Information-theoretic setting, adversaries must be optimal, queries are bounded,. . .)

GCM has no Multi-Key Degradation



GCM has no Multi-Key Degradation



Summary

1. Multi-Key setting is important

Summary

1. Multi-Key setting is important
2. Generic attacks against block ciphers, but not against modes

Summary

1. Multi-Key setting is important
2. Generic attacks against block ciphers, but not against modes
3. Weak key attack

Summary

1. Multi-Key setting is important
2. Generic attacks against block ciphers, but not against modes
3. Weak key attack
4. Characterization of multi-key setting

Summary

1. Multi-Key setting is important
2. Generic attacks against block ciphers, but not against modes
3. Weak key attack
4. Characterization of multi-key setting

Open problem: ideal primitive settings?

Summary

1. Multi-Key setting is important
2. Generic attacks against block ciphers, but not against modes
3. Weak key attack
4. Characterization of multi-key setting

Open problem: ideal primitive settings?

Thank you for your attention.