Multi-Instance Security and its Application to Password-Based Cryptography

Stefano Tessaro

MIT

Joint work with
Mihir Bellare (UC San Diego)
Thomas Ristenpart (Univ. of Wisconsin)

Scenario: File encryption

Want to store data in encrypted form using **symmetric encryption.**

Scenario: File encryption

Want to store data in encrypted form using **symmetric encryption.**

- Keys need to be **securely stored** for later decryption

Scenario: File encryption

Want to store data in encrypted form using **symmetric encryption.**

- Keys need to be **securely stored** for later decryption

Alternative solution:
Password-based cryptography.

Password-based encryption

Password-based encryption

Used widely: *Winzip, OpenOffice, Mac OS X FileVault,TrueCrypt, WiFi WPA (PBKDF), …*

Password-based encryption

Used widely: *Winzip, OpenOffice, Mac OS X FileVault,TrueCrypt, WiFi WPA (PBKDF), …*

$K = \mathbf{01001101} \ldots\ldots\ldots\ldots \mathbf{0101010101}$

Key-derivation function

KDF

q1w2e3

Password-based encryption

Used widely: *Winzip, OpenOffice, Mac OS X FileVault,TrueCrypt, WiFi WPA (PBKDF), …*

$\boldsymbol{K} = \mathbf{01001101 \ldots \ldots \ldots 0101010101}$

Key-derivation function

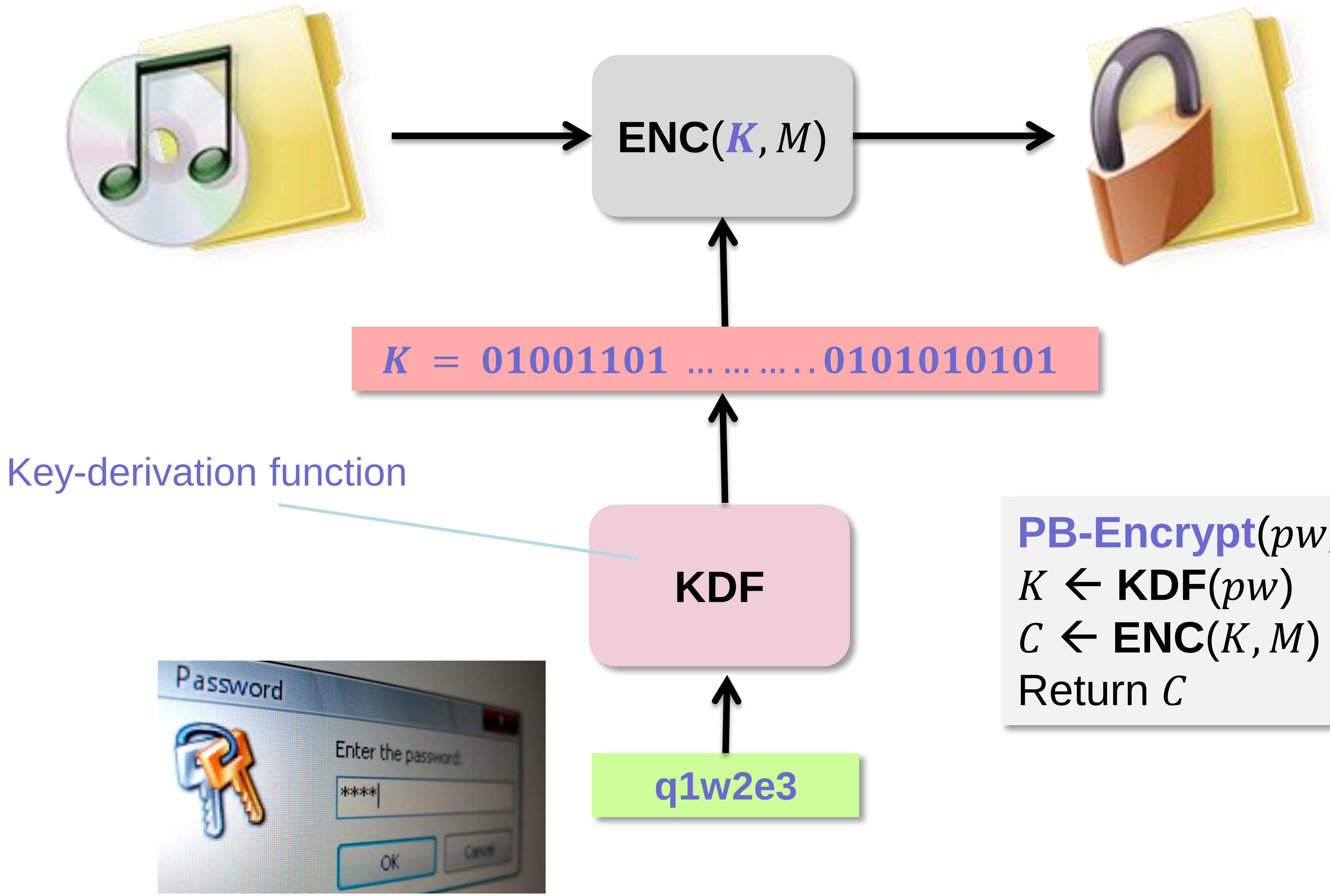
KDF

q1w2e3

PB-Encrypt(pw, M)
$K \leftarrow$ **KDF**(pw)
$C \leftarrow$ **ENC**(K, M)
Return C

Password-based encryption

Used widely: *Winzip, OpenOffice, Mac OS X FileVault,TrueCrypt, WiFi WPA (PBKDF), …*



PB-Encrypt(pw, M)
K ← **KDF**(pw)
C ← **ENC**(K, M)
Return C

Problem: Weak passwords are unavoidable

Problem: Weak passwords are unavoidable

Security on NBCNEWS.com

Breach shows even experts choose bad passwords

Easy-to-guess passwords such as '123456' are all too common

Jump to discuss
comments below

Below: Discuss Related

By Matt Liebowitz

securitynews DAILY

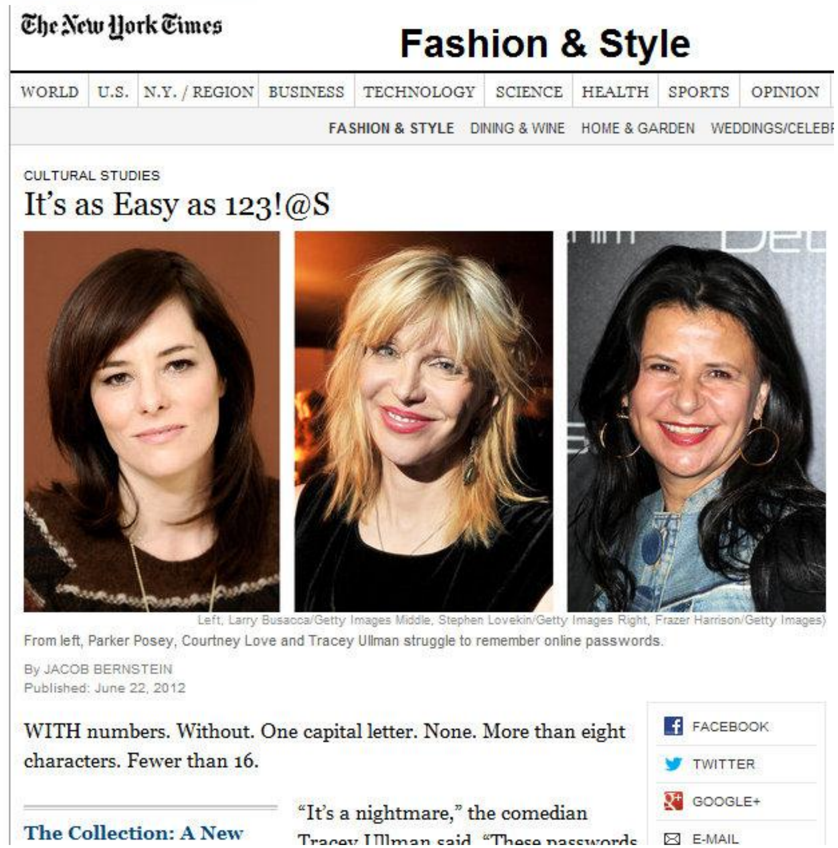
updated 1/4/2012 11:21:34 AM ET

Anonymous' massive year-end attack on the global-security consulting fi
even top-tier executives at the world's largest corporations don't have a cl
importance of a strong password.

Enter a password

abc35

Password Strength:



The New York Times

Fashion & Style

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

FASHION & STYLE | DINING & WINE | HOME & GARDEN | WEDDINGS/CELEBR

CULTURAL STUDIES

It's as Easy as 123!@S

Left, Larry Busacca/Getty Images Middle, Stephen Lovekin/Getty Images Right, Frazer Harrison/Getty Images)

From left, Parker Posey, Courtney Love and Tracey Ullman struggle to remember online passwords.

By JACOB BERNSTEIN
Published: June 22, 2012

FACEBOOK
TWITTER
GOOGLE+
E-MAIL

WITH numbers. Without. One capital letter. None. More than eight characters. Fewer than 16.

The Collection: A New

"It's a nightmare," the comedian
Tracey Ullman said. "These passwords

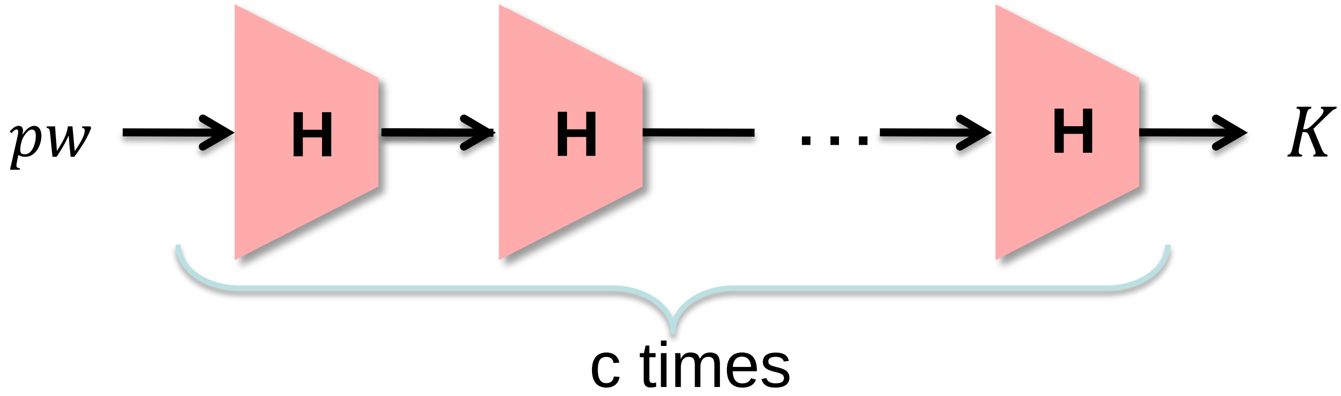
Problem: Weak passwords are unavoidable

Mitigating dictionary attacks via iteration

$\text{KDF} = \text{H}^c$

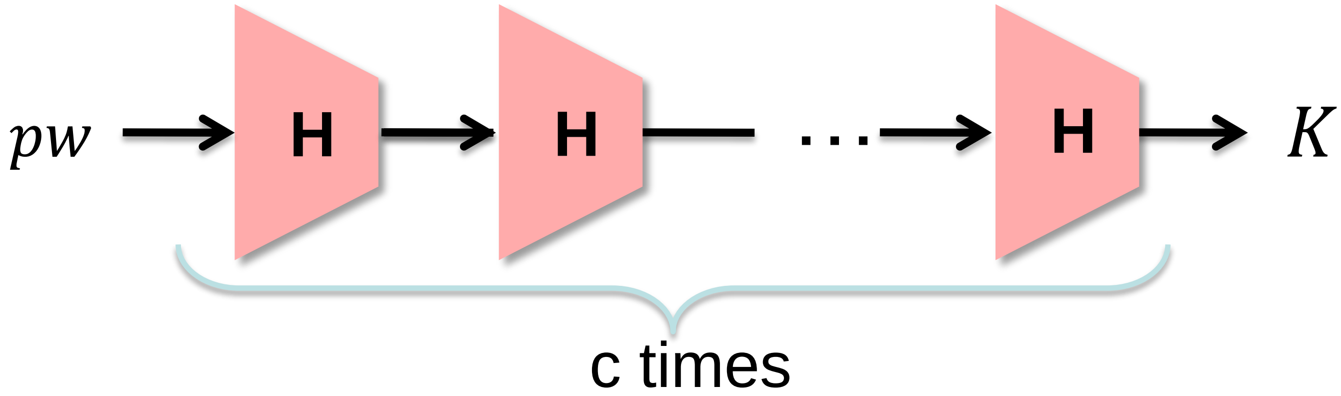
Mitigating dictionary attacks via iteration

KDF = H^c



Mitigating dictionary attacks via iteration

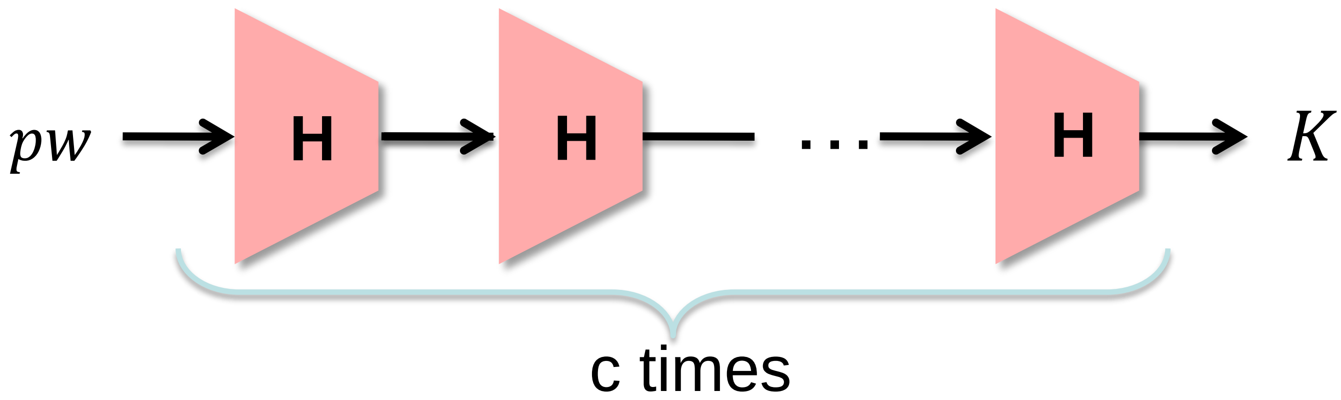
KDF = H^c



H : $\{0,1\}^* \rightarrow \{0,1\}^n$ is cryptographic hash function (e.g., SHA-256)

Mitigating dictionary attacks via iteration

KDF = H^c

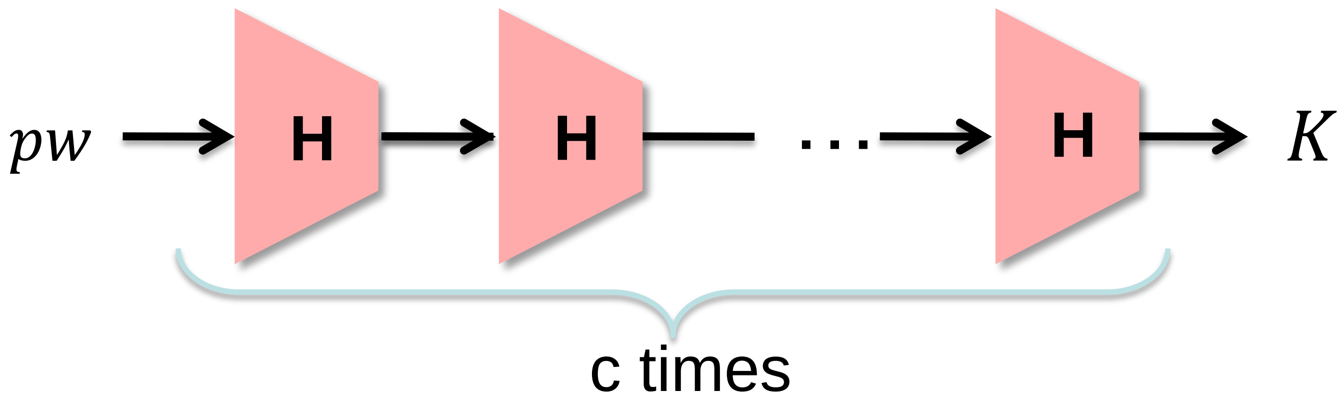


H : $\{0,1\}^* \rightarrow \{0,1\}^n$ is cryptographic hash function (e.g., SHA-256)

PB-Encrypt(pw, M)
$K \leftarrow$ **H**$^c(pw)$
$C \leftarrow$ **ENC**(K, M)
Return C

Mitigating dictionary attacks via iteration

KDF = H^c



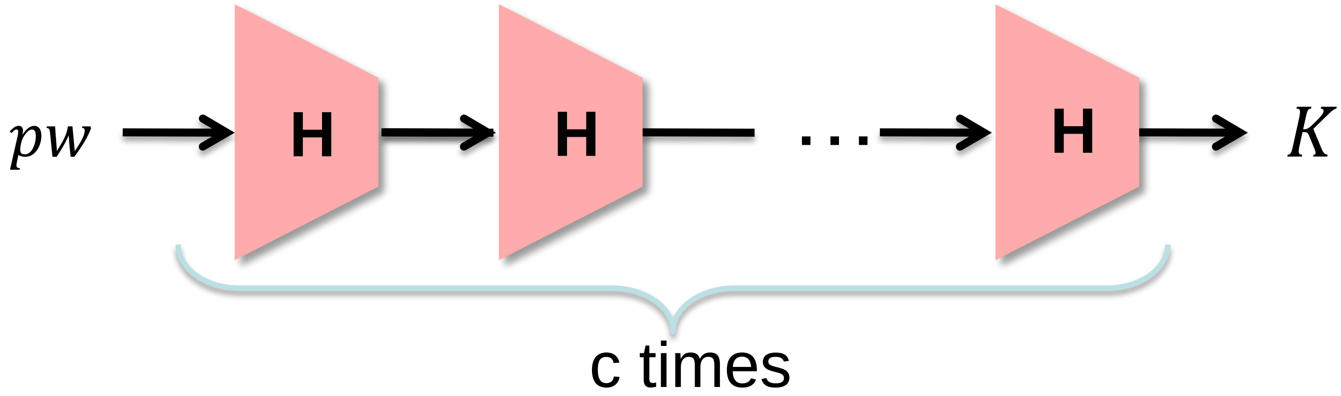
H : $\{0,1\}^* \rightarrow \{0,1\}^n$ is cryptographic hash function (e.g., SHA-256)

PB-Encrypt(pw, M)
K ← **H**$^c(pw)$
C ← **ENC**(K, M)
Return C

Expectation:
Work $\boldsymbol{N}$ to guess $pw \Rightarrow$ Work $\mathbf{c} \times \boldsymbol{N}$ to break **PB-Encrypt**

Mitigating dictionary attacks via iteration

KDF = H^c



H : $\{0,1\}^* \rightarrow \{0,1\}^n$ is cryptographic hash function (e.g., SHA-256)

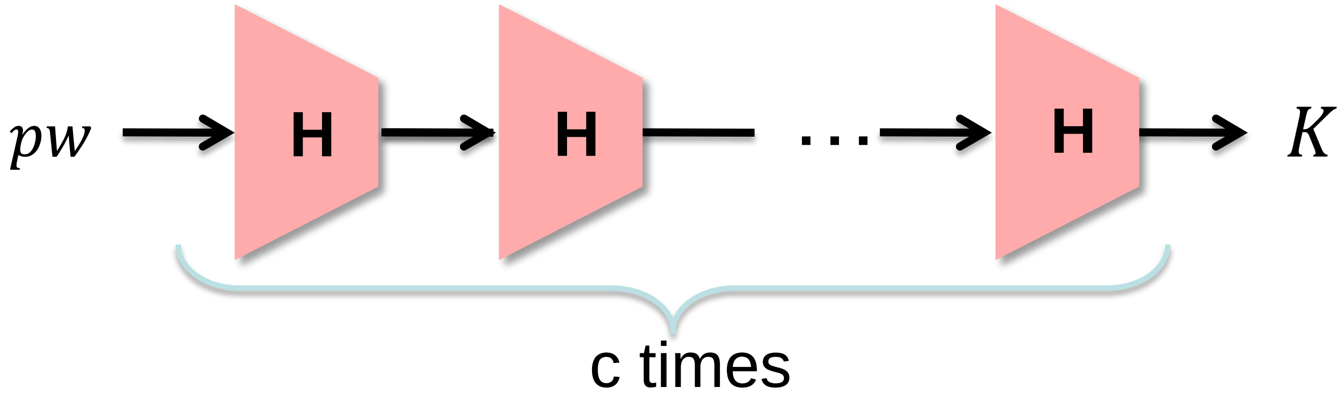
PB-Encrypt(pw, M)
$K \leftarrow$ **H**$^c(pw)$
$C \leftarrow$ **ENC**(K, M)
Return C

Expectation:
Work $\boldsymbol{N}$ to guess $pw \Rightarrow$ Work $\mathbf{c} \times \boldsymbol{N}$ to break **PB-Encrypt**

$$N = 2^{32}$$

Mitigating dictionary attacks via iteration

KDF = H^c



H : $\{0,1\}^* \rightarrow \{0,1\}^n$ is cryptographic hash function (e.g., SHA-256)

PB-Encrypt(pw, M)
$K \leftarrow$ **H**$^c(pw)$
$C \leftarrow$ **ENC**(K, M)
Return C

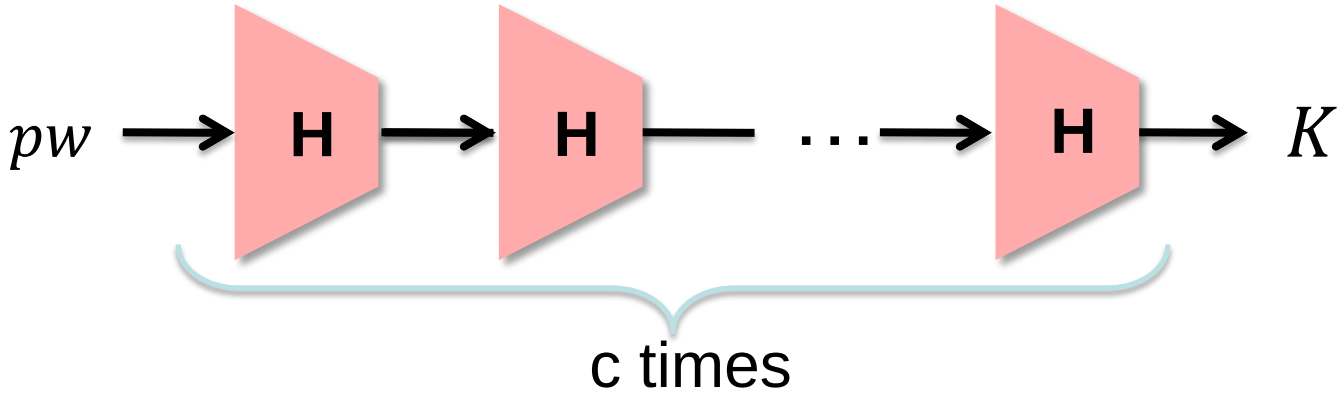
Expectation:
Work N to guess $pw \Rightarrow$ Work $\mathbf{c} \times N$ to break **PB-Encrypt**

$$N = 2^{32}$$

$$N \times c = 2^{32} \times 2^{20} = 2^{52}$$

Mitigating dictionary attacks via iteration

KDF = H^c



H : $\{0,1\}^* \rightarrow \{0,1\}^n$ is cryptographic hash function (e.g., SHA-256)

PB-Encrypt(pw, M)
$K \leftarrow$ **H**$^c(pw)$
$C \leftarrow$ **ENC**(K, M)
Return C

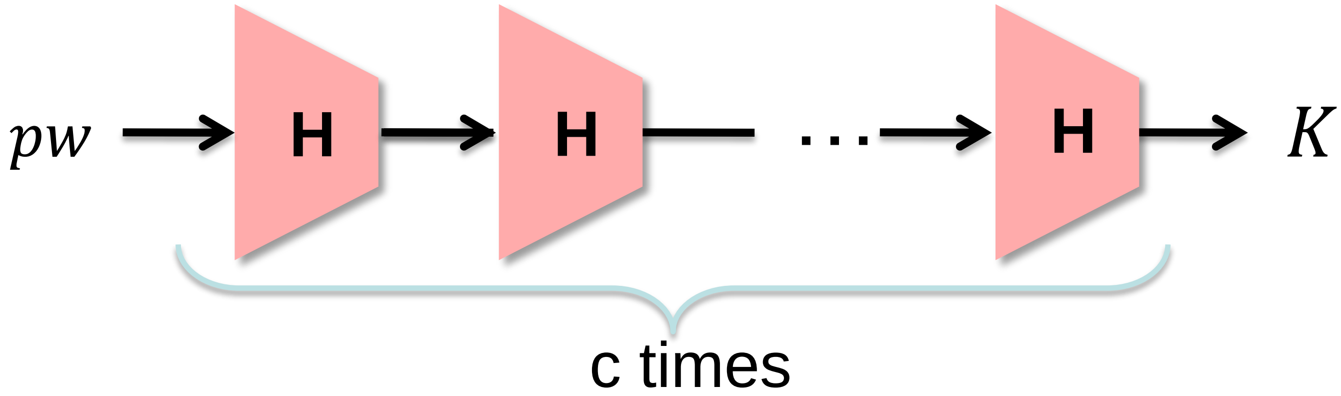
Expectation:
Work $\boldsymbol{N}$ to guess $pw \Rightarrow$ Work $\mathbf{c} \times \boldsymbol{N}$ to break **PB-Encrypt**

$$N = 2^{32}$$

$$N \times c = 2^{32} \times 2^{20} = 2^{52}$$

Mitigating dictionary attacks via iteration

KDF = H^c



H : $\{0,1\}^* \rightarrow \{0,1\}^n$ is cryptographic hash function (e.g., SHA-256)

PB-Encrypt(pw, M)
$K \leftarrow$ **H**$^c(pw)$
$C \leftarrow$ **ENC**(K, M)
Return C

Expectation:
Work N to guess pw $\Rightarrow$ Work $c \times N$ to break **PB-Encrypt**

$N = 2^{32}$

$N \times c = 2^{32} \times 2^{20} = 2^{52}$

PB-Encryption in the multi-user setting

Real world has multiple users:

PB-Encryption in the multi-user setting

Real world has multiple users:

$C_1 \leftarrow$ **PB−Encrypt**(pw_1, M_1)

$C_2 \leftarrow$ **PB−Encrypt**(pw_2, M_2)

$C_3 \leftarrow$ **PB−Encrypt**(pw_3, M_3)

PB-Encryption in the multi-user setting

Real world has multiple users:

$C_1 \leftarrow$ **PB−Encrypt**(pw_1, M_1)

$C_2 \leftarrow$ **PB−Encrypt**(pw_2, M_2)

$C_3 \leftarrow$ **PB−Encrypt**(pw_3, M_3)

PB-Encryption in the multi-user setting

Real world has multiple users:

$C_1 \leftarrow$ **PB−Encrypt**(pw_1, M_1)

$C_2 \leftarrow$ **PB−Encrypt**(pw_2, M_2)

$C_3 \leftarrow$ **PB−Encrypt**(pw_3, M_3)

PB-Encryption in the multi-user setting

Real world has multiple users:

$C_1 \leftarrow$ **PB−Encrypt**(pw_1, M_1)

$C_2 \leftarrow$ **PB−Encrypt**(pw_2, M_2)

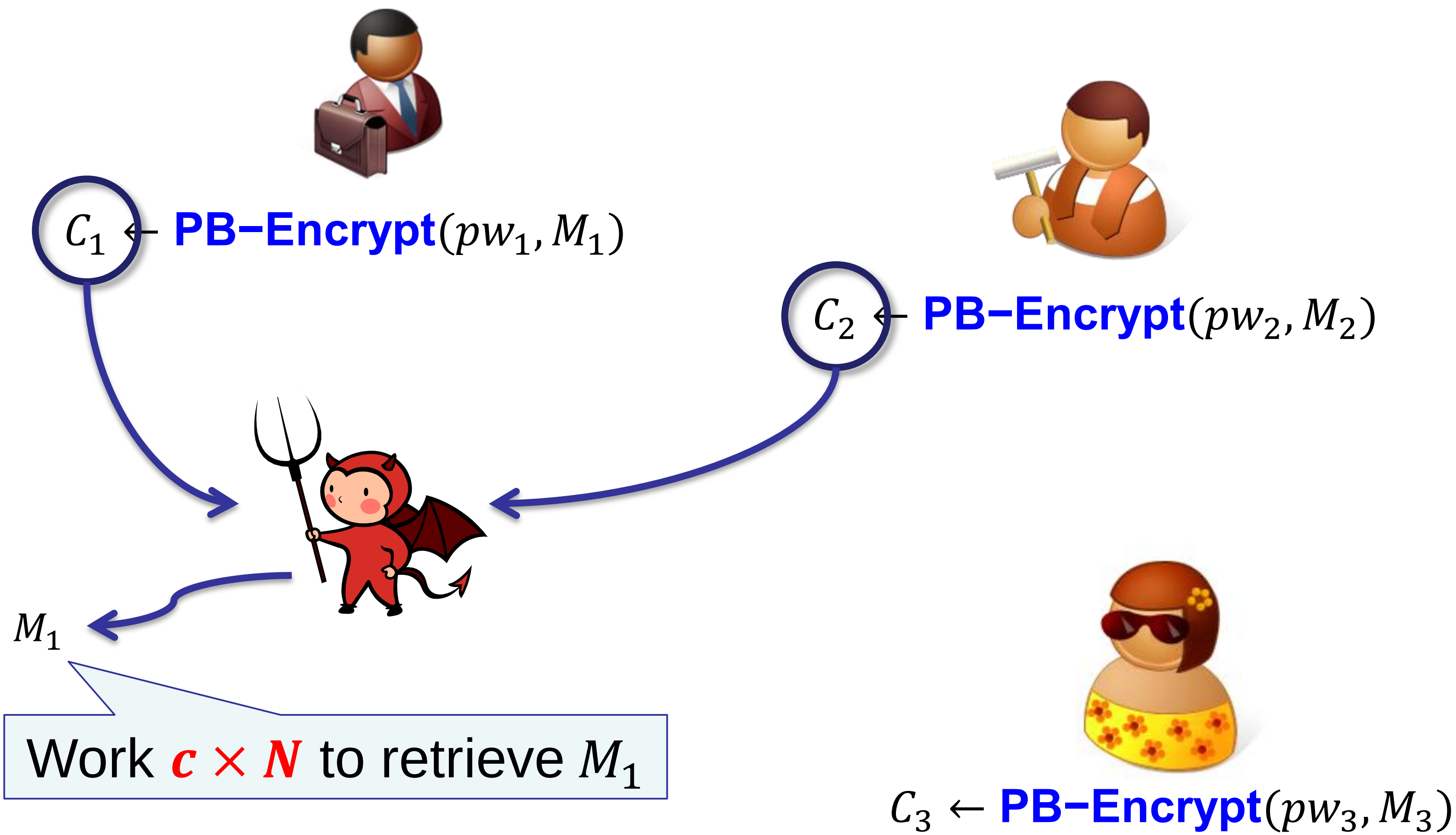
M_1

Work $\boldsymbol{c} \times \boldsymbol{N}$ to retrieve M_1

$C_3 \leftarrow$ **PB−Encrypt**(pw_3, M_3)

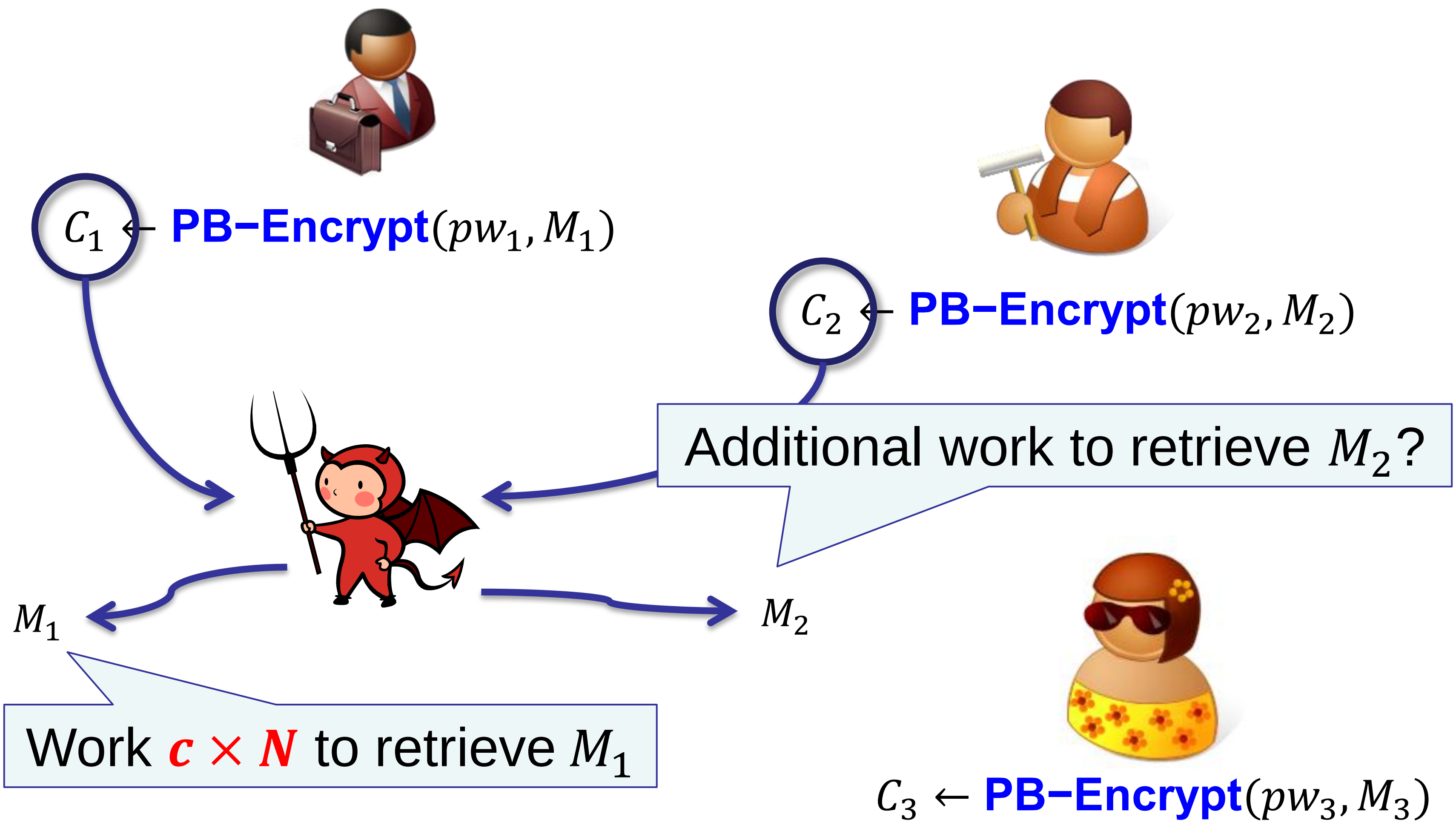
PB-Encryption in the multi-user setting

Real world has multiple users:



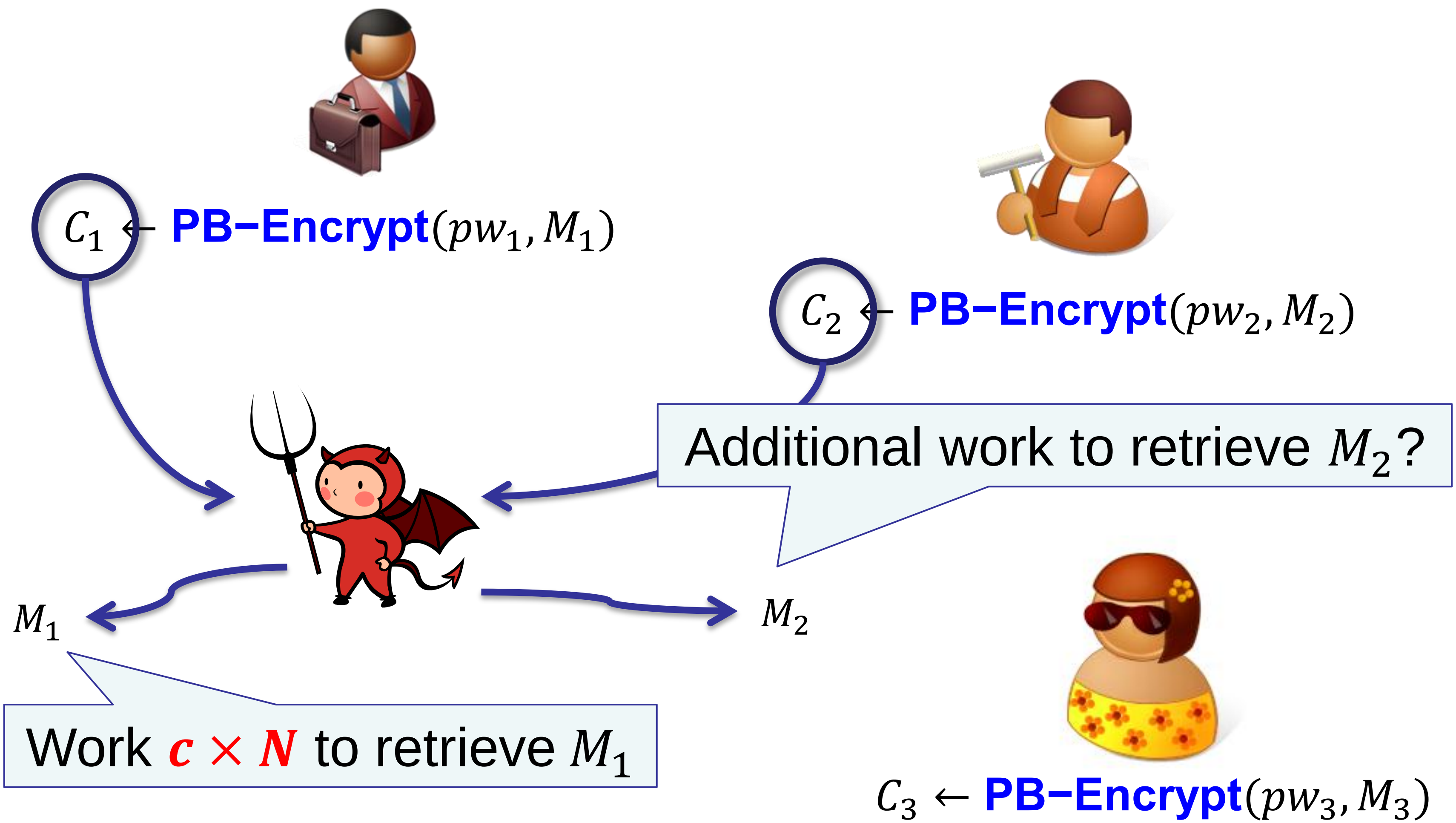
PB-Encryption in the multi-user setting

Real world has multiple users:



PB-Encryption in the multi-user setting

Real world has multiple users:



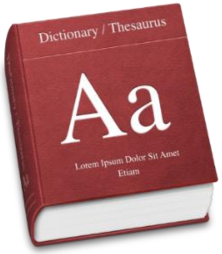
Ideally: Work $m \times c \times N$ to retrieve m plaintexts!

Multi-instance security amplification

Not true in general:

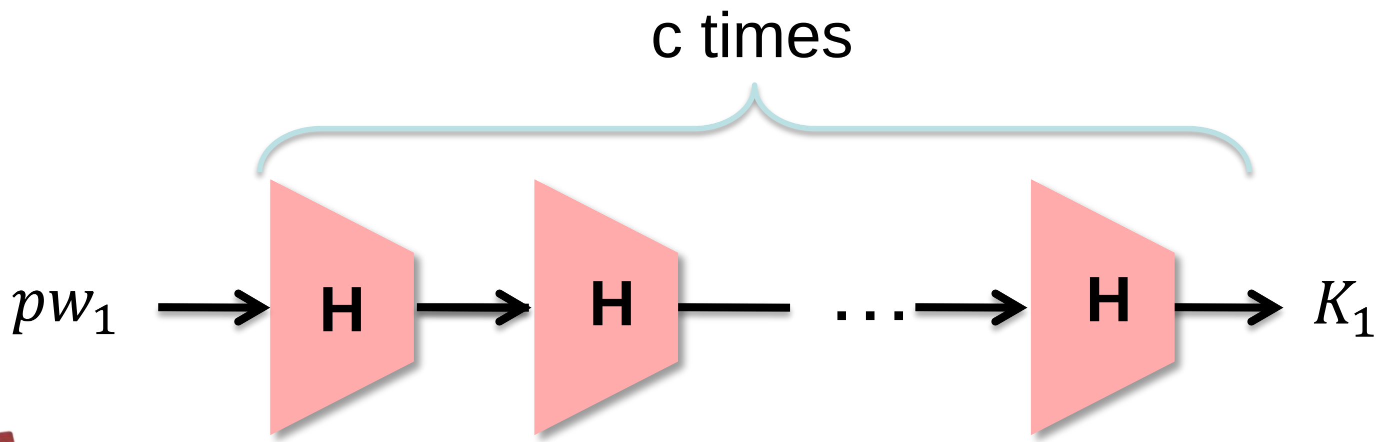
Multi-instance security amplification

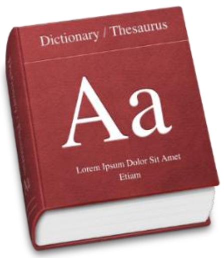
Not true in general:



Multi-instance security amplification

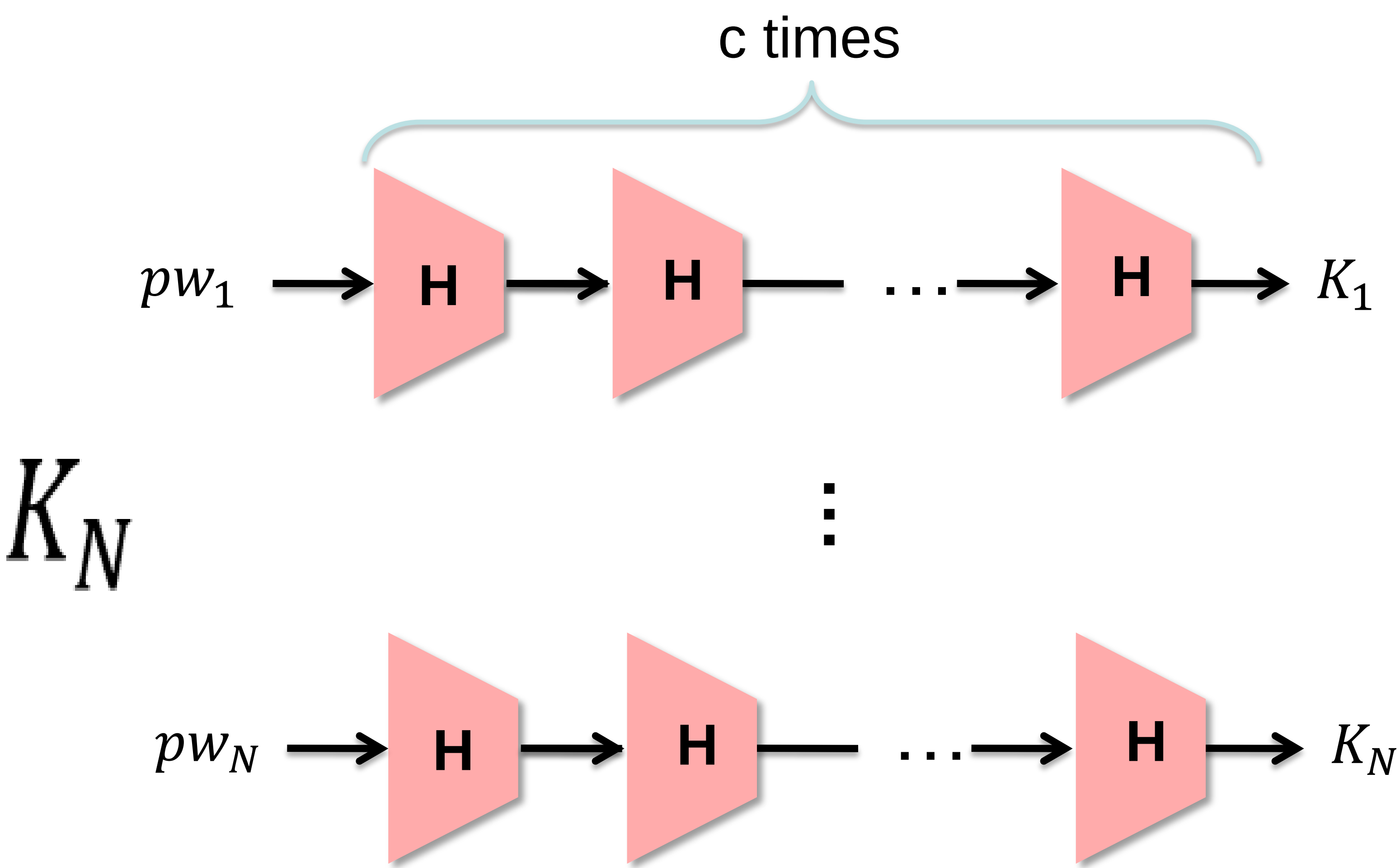
Not true in general:





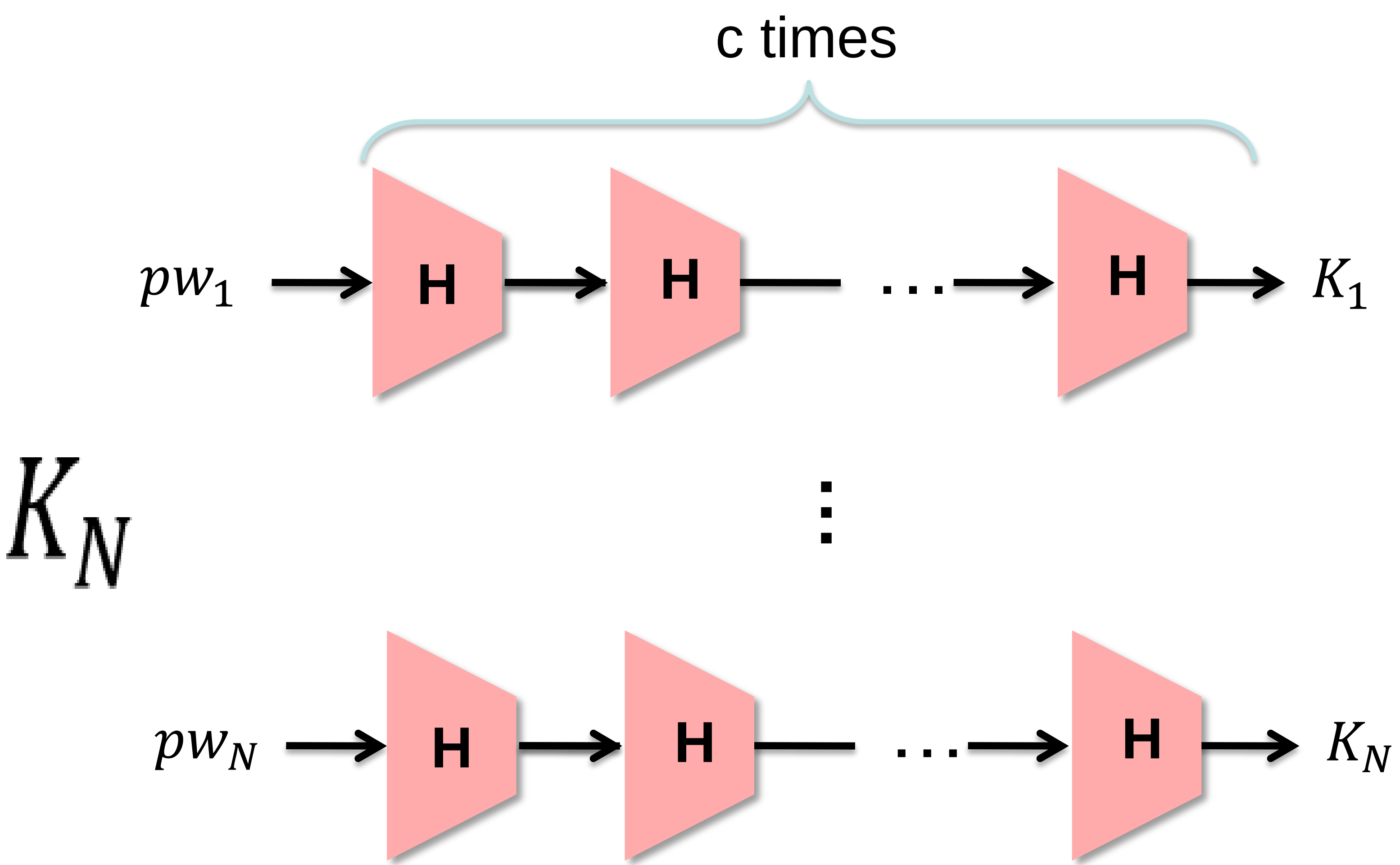
Multi-instance security amplification

Not true in general:



Multi-instance security amplification

Not true in general:

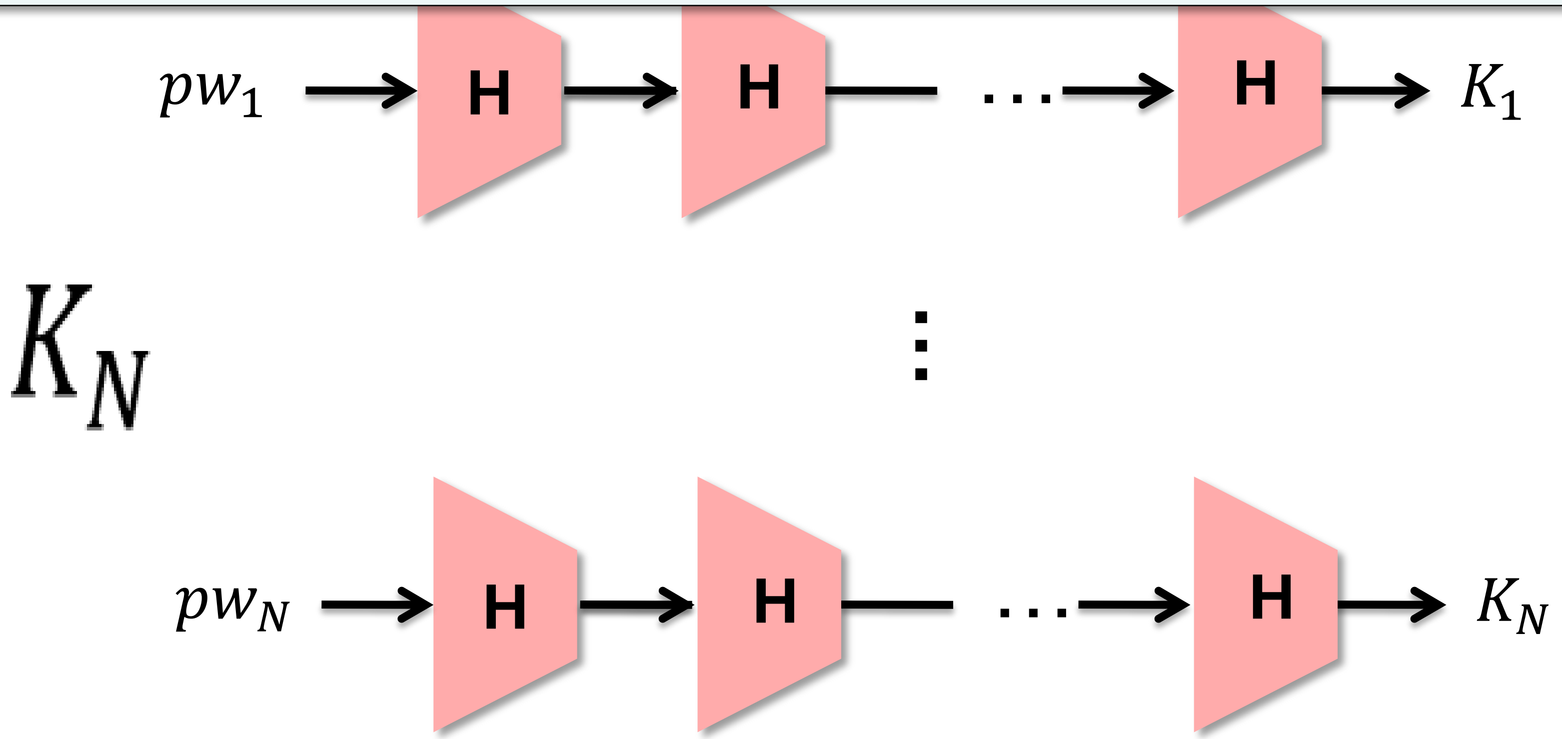


Work $N \times c$ + Work N / ciphertext = $N \times (c + m)$ vs $N \times c \times m$

Multi-instance security amplification

Not true in general:

New design goal: Multi-instance security amplification
"Hardness of breaking multiple instances must increase linearly in the number of instances."



Work $N \times c$ + Work N / ciphertext = $N \times (c + m)$ vs $N \times c \times m$

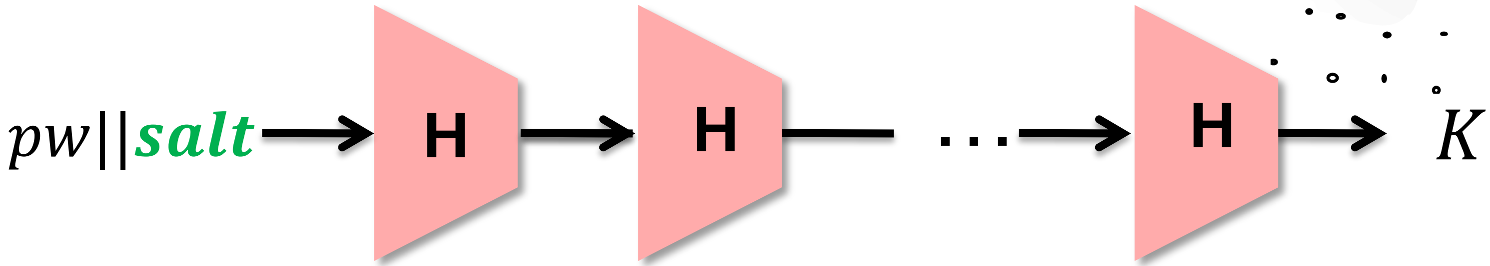
PKCS#5 – Password-based cryptography standard

Salting as suggested in PKCS#5 prevents attack

PKCS#5 – Password-based cryptography standard

Salting as suggested in PKCS#5 prevents attack

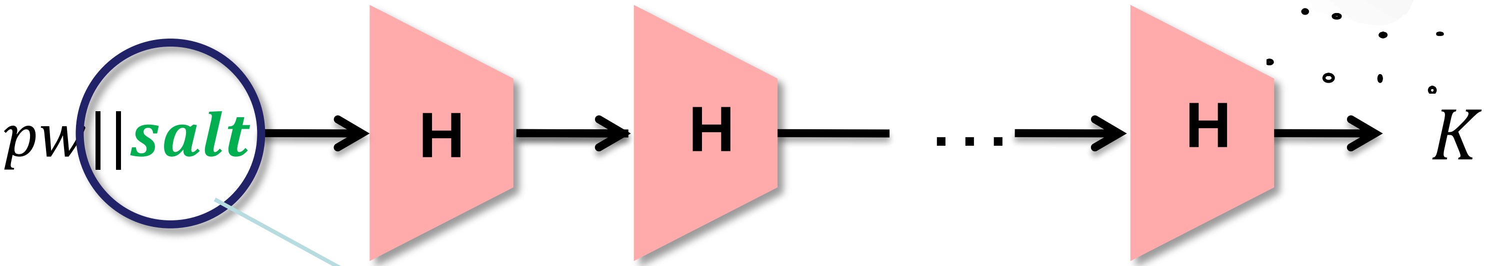
KDF1:



PKCS#5 – Password-based cryptography standard

Salting as suggested in PKCS#5 prevents attack

KDF1:

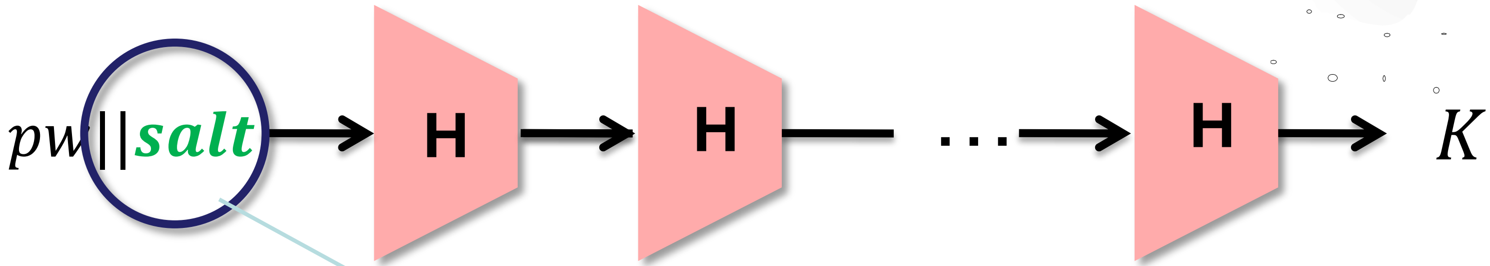


Randomly chosen per KDF evaluation

PKCS#5 – Password-based cryptography standard

Salting as suggested in PKCS#5 prevents attack

KDF1:



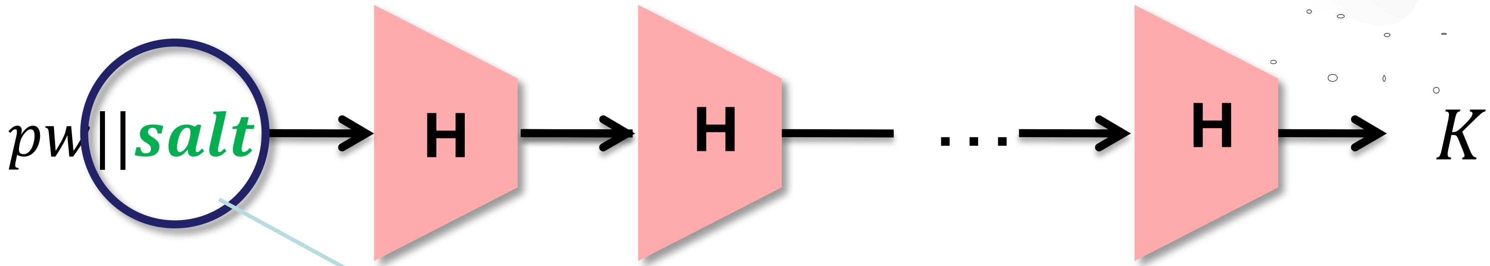
Randomly chosen per KDF evaluation

PB-Encrypt(pw, M)
$\boldsymbol{salt} \leftarrow \{0,1\}^s$
$K \leftarrow \mathbf{H}^c(pw||\boldsymbol{salt})$
$C \leftarrow \mathbf{ENC}(K, M)$
Return $C||\boldsymbol{salt}$

PKCS#5 – Password-based cryptography standard

Salting as suggested in PKCS#5 prevents attack

KDF1:



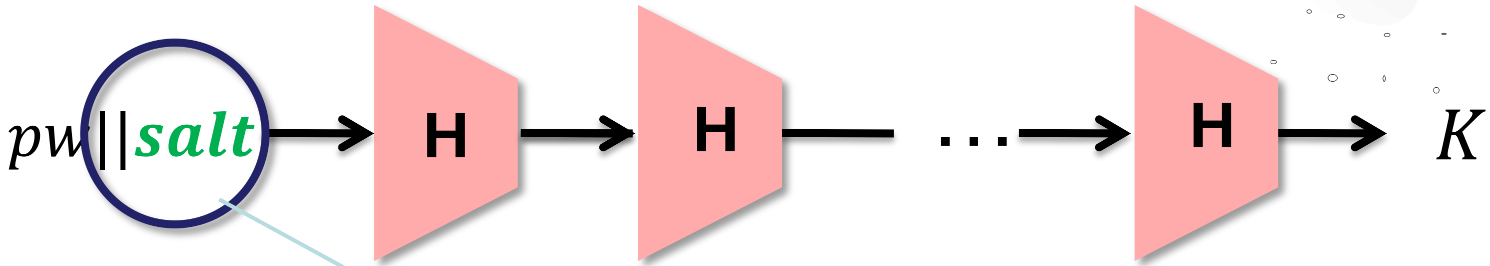
PB-Encrypt(pw, M)
$\boldsymbol{salt} \leftarrow \{0,1\}^s$
$K \leftarrow \mathbf{H}^c(pw||\boldsymbol{salt})$
$C \leftarrow \mathbf{ENC}(K, M)$
Return $C||\boldsymbol{salt}$

Randomly chosen per KDF evaluation

PKCS#5 – Password-based cryptography standard

Salting as suggested in PKCS#5 prevents attack

KDF1:



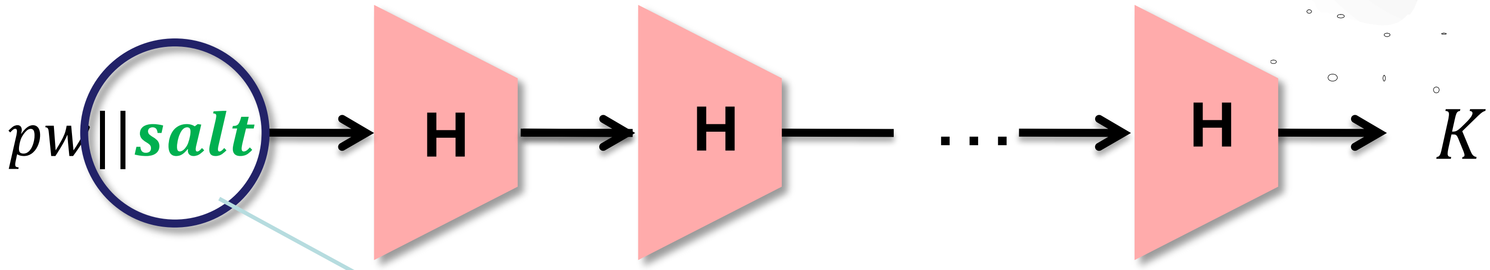
Randomly chosen per KDF evaluation

PB-Encrypt(pw, M)
$salt \leftarrow \{0,1\}^s$
$K \leftarrow \mathbf{H}^c(pw||salt)$
$C \leftarrow \mathbf{ENC}(K, M)$
Return $C||salt$

PKCS#5 – Password-based cryptography standard

Salting as suggested in PKCS#5 prevents attack

KDF1:



PB-Encrypt(pw, M)
$salt \leftarrow \{0,1\}^s$
$K \leftarrow \mathbf{H}^c(pw||salt)$
$C \leftarrow \mathbf{ENC}(K, M)$
Return $C||salt$

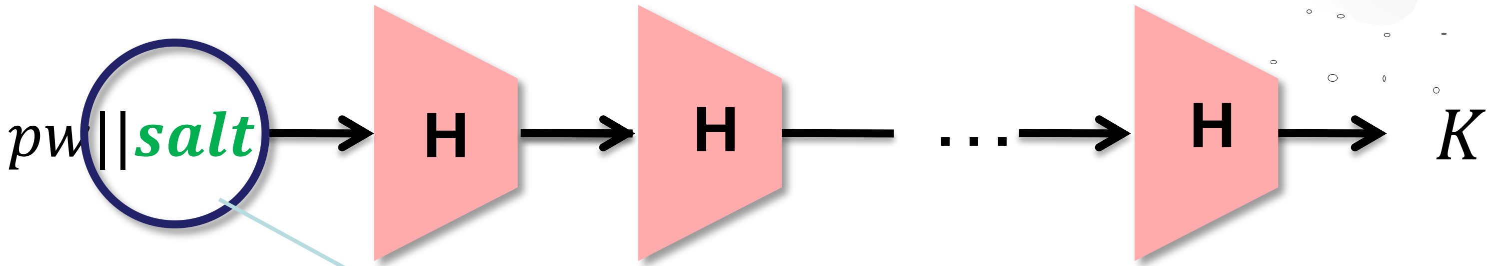
Randomly chosen per KDF evaluation

Allows decryption

PKCS#5 – Password-based cryptography standard

Salting as suggested in PKCS#5 prevents attack

KDF1:



PB-Encrypt(pw, M)
$salt \leftarrow \{0,1\}^s$
$K \leftarrow$ **H**$^c(pw||salt)$
$C \leftarrow$ **ENC**(K, M)
Return $C||salt$

Randomly chosen per KDF evaluation

Allows decryption

Question: Does salting provably ensure multi-instance security amplification?

Iteration and salting in the real world

ZDNet

White Papers Hot Topics Downloads Reviews Newsletters

US Edition Data Centers Olympics 2012 Patents Apple Microsoft Security

READ THIS: *Is 'Metro' now a banned word at Microso*

Topic: *Security* *Follow via:*

6.46 million LinkedIn passwords leaked online

Summary: *More than 6.4 million LinkedIn pa... hack. Though some login details are encrypted,*

By Zack Whittaker for Between the Lin...

Follow @zackwhittaker

A user on a Russian forum has claimed to have ... from LinkedIn.

It looks as though some of the weaker passwo... cracked already. Other users have been seen r... seek help in cracking the encryption.

No salting!

acmqueue

Association for Computing Machinery

A Special Offer to Join ACM Why Join A...

HOME AUDIOCASTS VI...

COLUMNS > THE BIKESHED

LinkedIn Password Leak: Salt Their Hide

view issue

by Poul-Henning Kamp | June 7, 2012

Topic: Security

Like 196

Site Feeds
Queue Blog
Twitter feed
Facebook

BROWSE TOPICS
Databases
Patching and Deployment
Programming Languages
Web Development
Networks
Performance

If it does not take a full second to calculate the password hash, it is too weak.

POUL-HENNING KAMP

6.5 million unsalted SHA1 hashed LinkedIn passwords have appeared in the criminal underground. There are two words in that sentence that should cause LinkedIn no end of concern: "unsalted" and "SHA1."

No iteration!

Our results

Our results

Question: Does salting provably ensure multi-instance security amplification?

Our results

Question: Does salting provably ensure multi-instance security amplification?

Answer: We do not really know!

Our results

Question: Does salting provably ensure multi-instance security amplification?

Answer: We do not really know!

1) **No formal proof!**

Our results

Question: Does salting provably ensure multi-instance security amplification?

Answer: We do not really know!

1) **No formal proof!**
2) **No formal model!**

Our results

Question: Does salting provably ensure multi-instance security amplification?

Answer: We do not really know!

1) **No formal proof!**
2) **No formal model!**

Our contributions:

1) **General definitional framework for multi-instance security** of arbitrary cryptographic primitives.
2) **Case study:** Security analysis of PKCS#5 within our framework.

Outline

1. Multi-instance security

2. Security of PKCS#5 – A case study

Outline

1. Multi-instance security

2. Security of PKCS#5 – A case study

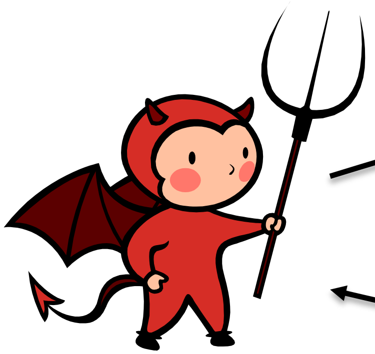
Single-instance security – PB-Encryption

LOR-Security

$b \leftarrow \{0,1\}$
$pw \leftarrow PWD$

Single-instance security – PB-Encryption

LOR-Security



m_0, m_1

$|m_0| = |m_1|$

$b \leftarrow \{0,1\}$

$pw \leftarrow PWD$

$\mathrm{ENC}(pw, m_b)$

Single-instance security – PB-Encryption

<u>**LOR-Security**</u>

m_0, m_1

$|m_0| = |m_1|$

$b \leftarrow \{0,1\}$

$pw \leftarrow PWD$

$\mathrm{ENC}(pw, m_b)$

b'

Single-instance security – PB-Encryption

LOR-Security

m_0, m_1

$|m_0| = |m_1|$

$b \leftarrow \{0,1\}$

$pw \leftarrow PWD$

$\mathrm{ENC}(pw, m_b)$

b'

$$\mathbf{Adv}^{\mathrm{lor}}(A) = 2 \times [\Pr[b = b'] - 1/2]$$

Single-instance security – PB-Encryption

LOR-Security

$\boldsymbol{m_0, m_1}$

$|\boldsymbol{m_0}| = |\boldsymbol{m_1}|$

$b \leftarrow \{0,1\}$

$pw \leftarrow PWD$

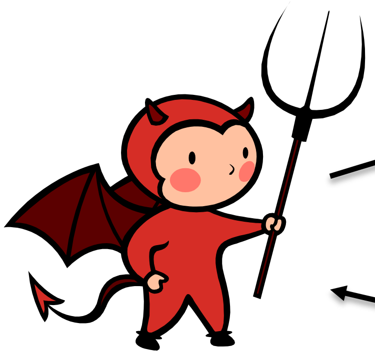
$\mathbf{ENC}(\boldsymbol{pw}, \boldsymbol{m_b})$

$\boldsymbol{b'}$

$$\boxed{\mathbf{Adv}^{\mathrm{lor}}(A) = 2 \times [\Pr[\boldsymbol{b} = \boldsymbol{b'}] - 1/2]}$$

Single-instance security – PB-Encryption

LOR-Security



m_0, m_1

$|m_0| = |m_1|$

$b \leftarrow \{0,1\}$

$pw \leftarrow PWD$

$\mathrm{ENC}(pw, m_b)$

b'

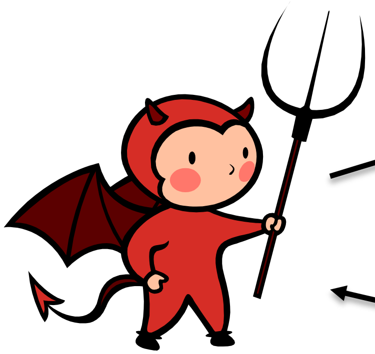
$$\boxed{\mathbf{Adv}^{\mathrm{lor}}(A) = 2 \times [\Pr[b = b'] - 1/2]}$$

PWR-Security

$pw \leftarrow PWD$

Single-instance security – PB-Encryption

LOR-Security



$\boldsymbol{m_0, m_1}$

$|\boldsymbol{m_0}| = |\boldsymbol{m_1}|$

$b \leftarrow \{0,1\}$

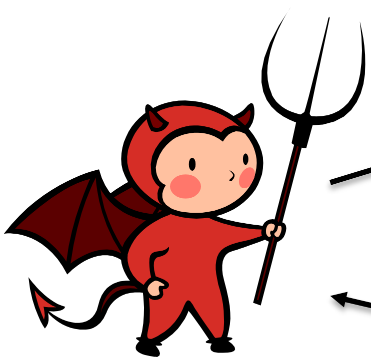
$pw \leftarrow PWD$

$\mathbf{ENC}(\boldsymbol{pw}, \boldsymbol{m_b})$

$\boldsymbol{b'}$

$$\boxed{\mathbf{Adv}^{\mathrm{lor}}(A) = 2 \times [\Pr[\boldsymbol{b} = \boldsymbol{b'}] - 1/2]}$$

PWR-Security



$\boldsymbol{m}$

$pw \leftarrow PWD$

$\mathbf{ENC}(\boldsymbol{pw}, \boldsymbol{m})$

Single-instance security – PB-Encryption

LOR-Security

m_0, m_1

$|m_0| = |m_1|$

$b \leftarrow \{0,1\}$

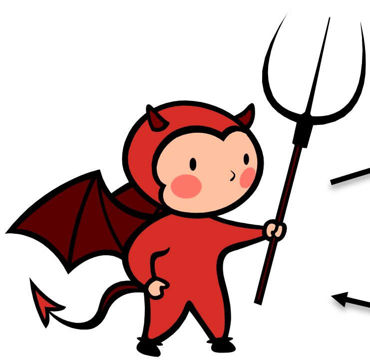
$pw \leftarrow PWD$

$\mathbf{ENC}(pw, m_b)$

b'

$$\boxed{\mathbf{Adv}^{\mathrm{lor}}(A) = 2 \times [\Pr[b = b'] - 1/2]}$$

PWR-Security



m

$pw \leftarrow PWD$

$\mathbf{ENC}(pw, m)$

pw'

Single-instance security – PB-Encryption

LOR-Security

m_0, m_1

$|m_0| = |m_1|$

$b \leftarrow \{0,1\}$

$pw \leftarrow PWD$

$\mathrm{ENC}(pw, m_b)$

b'

$$\mathbf{Adv}^{\mathrm{lor}}(A) = 2 \times [\Pr[b = b'] - 1/2]$$

PWR-Security

m

$pw \leftarrow PWD$

$\mathrm{ENC}(pw, m)$

pw'

$$\mathbf{Adv}^{\mathrm{pwr}}(A) = \Pr[pw' = pw]$$

The multi-instance (mi) security vista

Our goal: Define security metric for scheme **S** wrt property **P** to measure success of an adversary that:

- instances of the scheme **concurrently**.
- **Corrupts** up to $t < m$ instances of the scheme (e.g., learns passwords).
- Wins if it **breaks P for all uncorrupted instances.**

The multi-instance (mi) security vista

Our goal: Define security metric for scheme **S** wrt property **P** to measure success of an adversary that:

- **Attacks** m instances of the scheme **concurrently**.
- **Corrupts** up to $t < m$ instances of the scheme (e.g., learns passwords).
- Wins if it **breaks P for all uncorrupted instances.**

The multi-instance (mi) security vista

< mm instances of the scheme (e.g., learns passwords).

Our **goal:** Define security metric for scheme **S** wrt property **P** to measure success of an adversary that:

- **Attacks** m instances of the scheme **concurrently**.
- **Corrupts** up to $t < m$ instances of the scheme (e.g., learns passwords).
- Wins if it **breaks P for all uncorrupted instances.**

The multi-instance (mi) security vista

< mm instances of the scheme (e.g., learns passwords). Our **goal:** Define security metric for scheme **S** wrt property **P** to measure success of an adversary that:

- **Attacks** m instances of the scheme **concurrently**.
- Wins if it **breaks P for all uncorrupted instances.**
- Wins if it **breaks P for all uncorrupted instances.**

PWR security

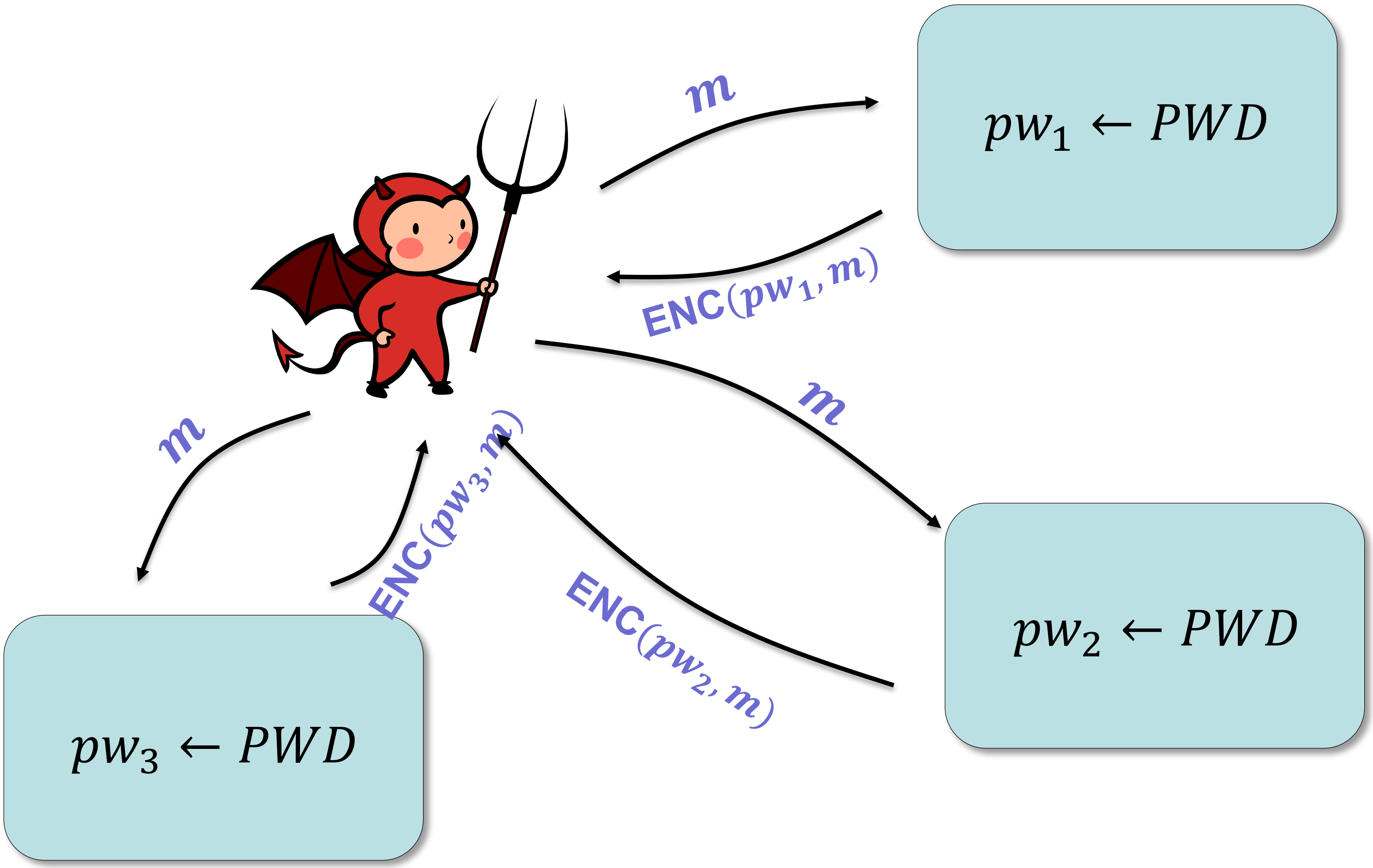
PWR security

$$pw_1 \leftarrow PWD$$

$$pw_2 \leftarrow PWD$$

$$pw_3 \leftarrow PWD$$

PWR security



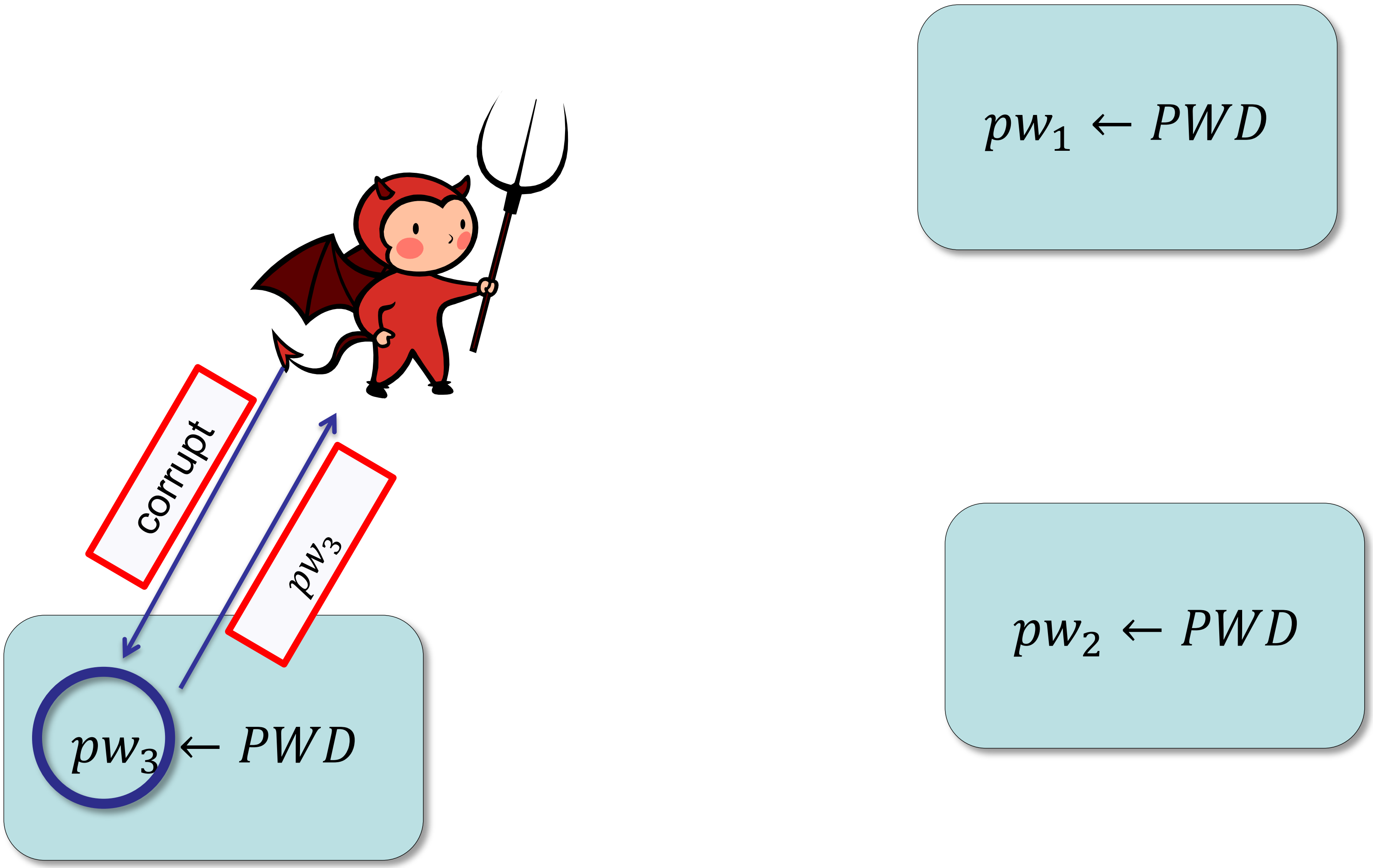
PWR security

$$pw_1 \leftarrow PWD$$

$$pw_2 \leftarrow PWD$$

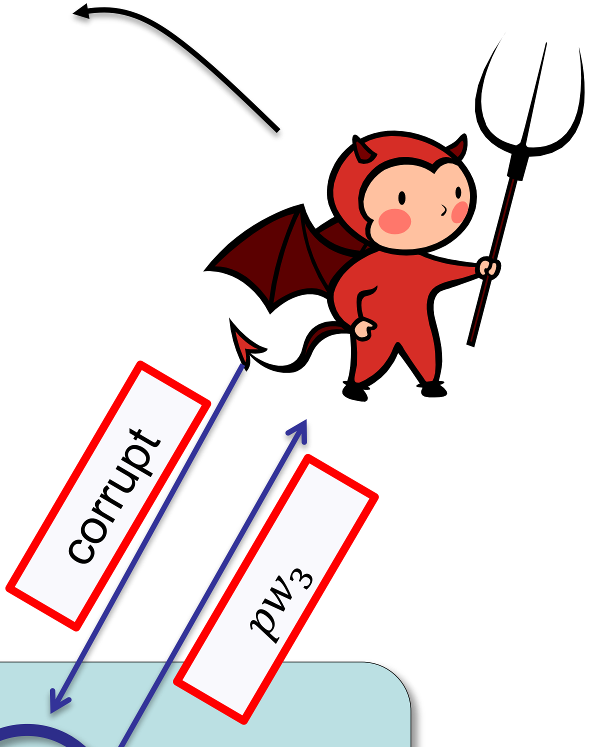
$$pw_3 \leftarrow PWD$$

PWR security



PWR security

(pw'_1, pw'_2, pw'_3)



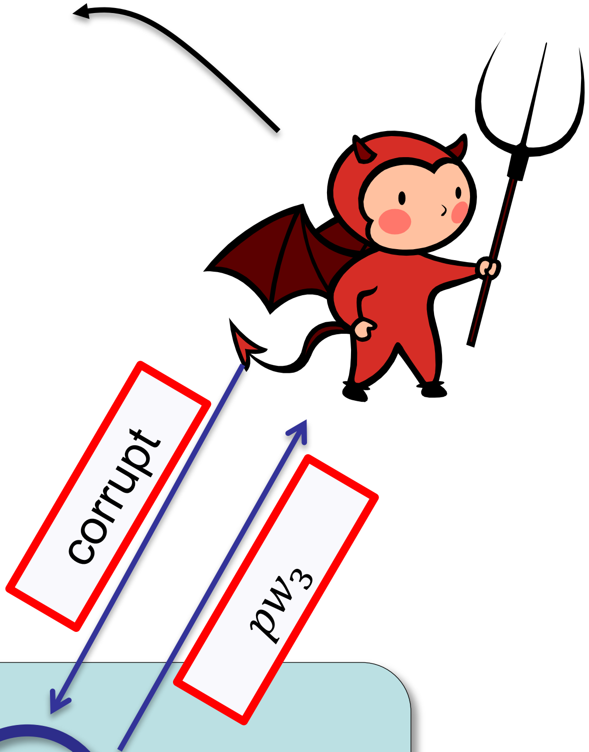
$pw_3 \leftarrow PWD$

$pw_1 \leftarrow PWD$

$pw_2 \leftarrow PWD$

PWR security

(pw'_1, pw'_2, pw'_3)



$pw_1 \leftarrow PWD$

$pw_2 \leftarrow PWD$

$pw_3 \leftarrow PWD$

$$\mathbf{Adv}^{\mathbf{m-pwr}}(A) = \Pr[pw'_1 = pw_1, \ldots, pw'_m = pw_m]$$

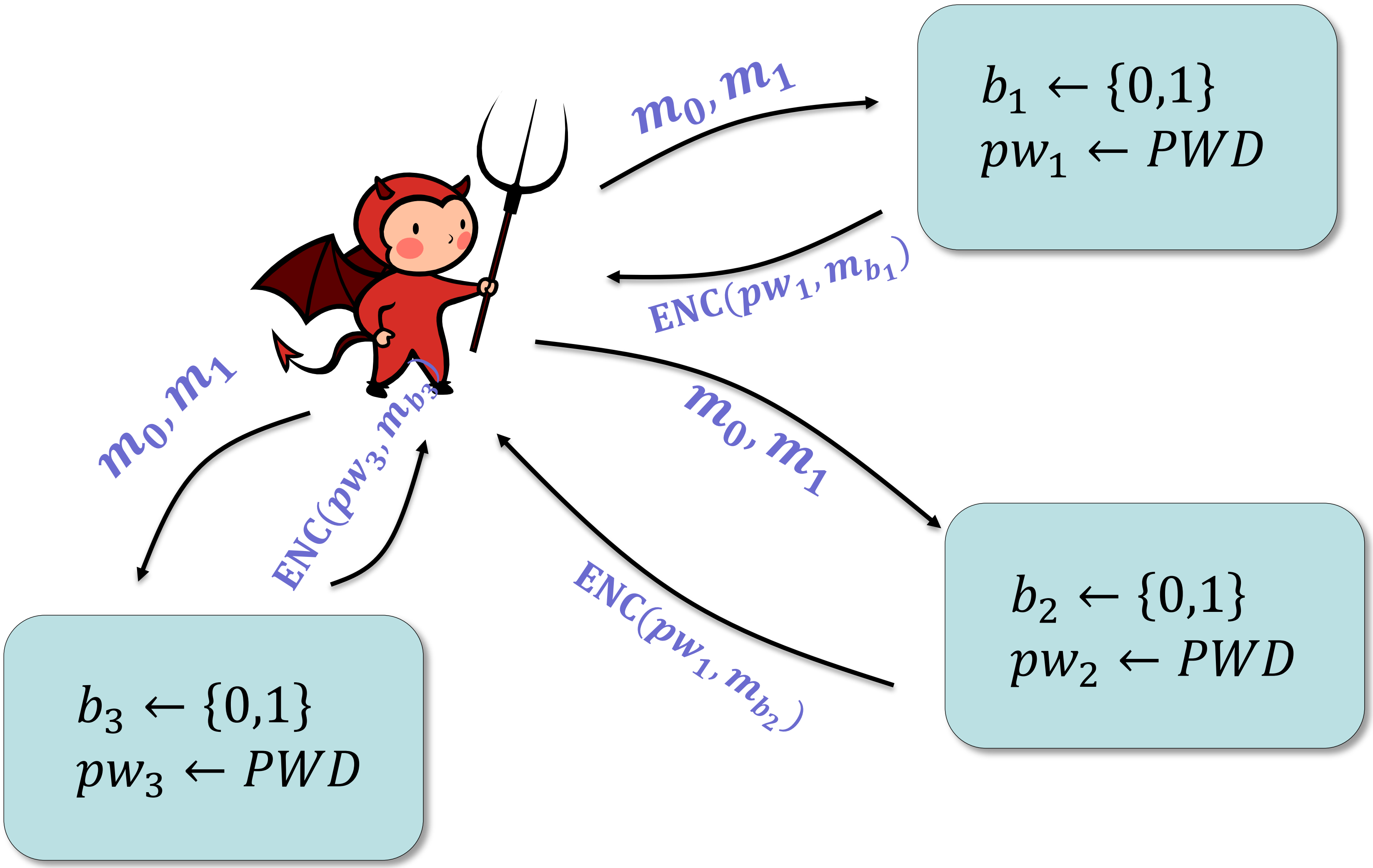
LOR security

$b_1 \leftarrow \{0,1\}$
$pw_1 \leftarrow PWD$

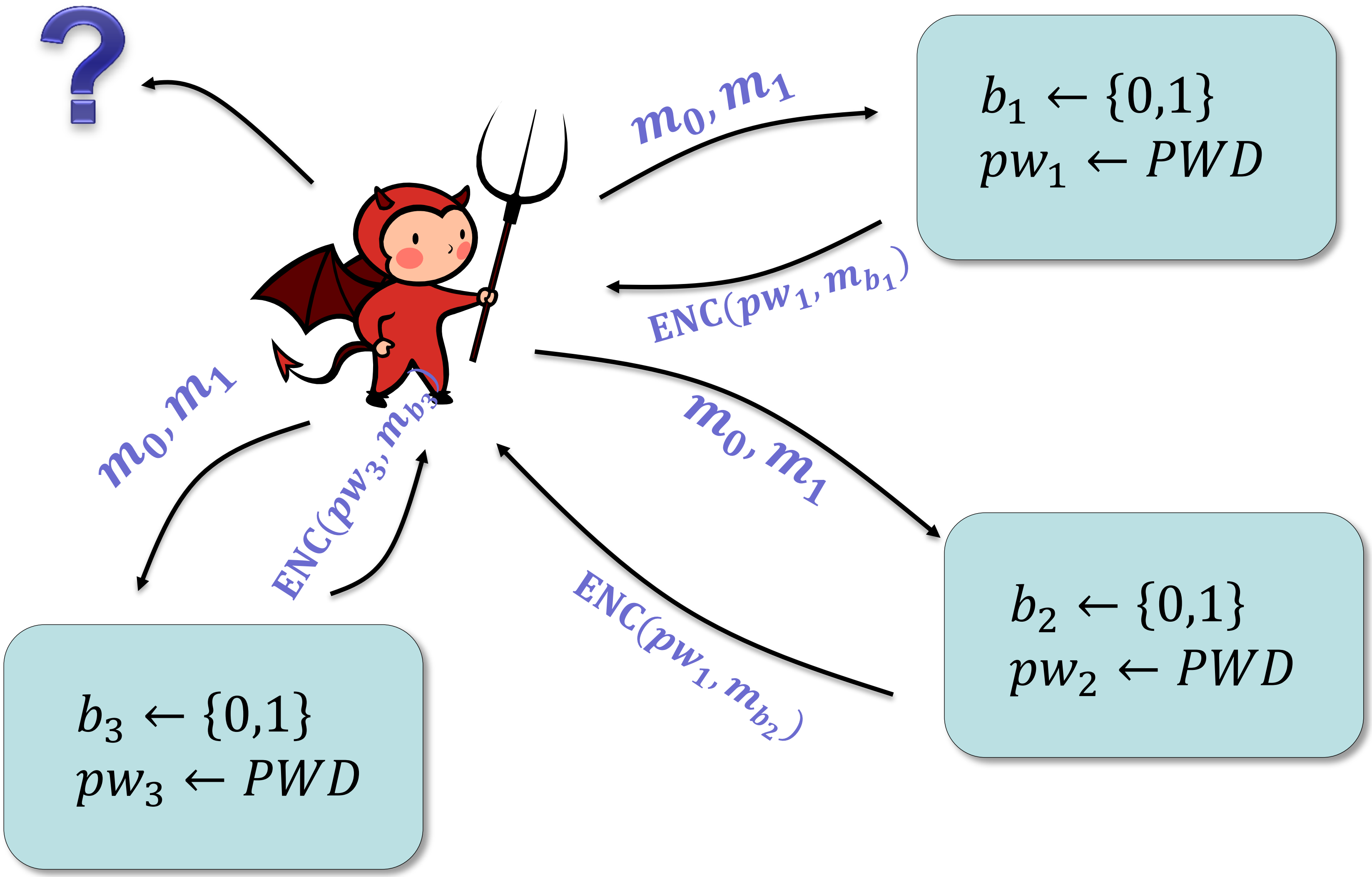
$b_2 \leftarrow \{0,1\}$
$pw_2 \leftarrow PWD$

$b_3 \leftarrow \{0,1\}$
$pw_3 \leftarrow PWD$

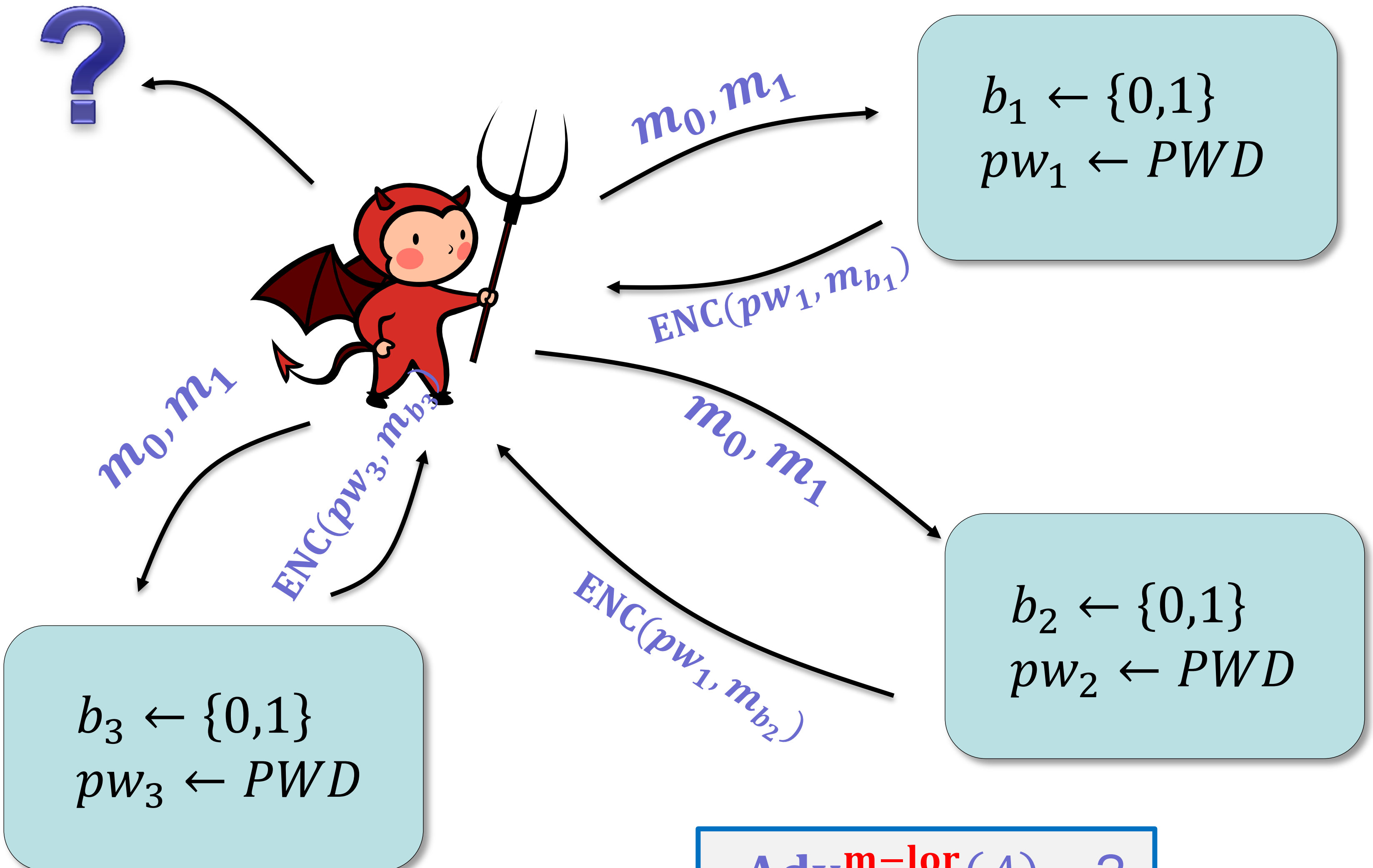
LOR security



LOR security



LOR security



$$\boxed{\mathbf{Adv}^{\mathbf{m-lor}}(A) = ?}$$

Defining mi security for encryption

Attempt #1: **AND-advantage**

Defining mi security for encryption

Attempt #1: **AND-advantage**

LORA-security: **Output:** $(b'_1, \ldots, b'_m)$

Advantage:

$$\mathbf{Adv}^{\mathbf{m-lora}}(A) = \mathbf{Pr}[(b_1, \ldots, b_m) = (b'_1, \ldots, b'_m)]$$

Defining mi security for encryption

Attempt #1: **AND-advantage**

LORA-security: **Output:** $(b'_1, \dots, b'_m)$

Advantage:

$$\mathbf{Adv}^{\mathrm{m-lora}}(A) = \mathbf{Pr}[(b_1, \dots, b_m) = (b'_1, \dots, b'_m)]$$

Problem: Does not measure hardness of winning all uncorrupted instances.

Defining mi security for encryption

Attempt #1: **AND-advantage**

LORA-security: **Output:** $(b'_1, \ldots, b'_m)$

Advantage:

$$\mathbf{Adv}^{\mathbf{m-lora}}(A) = \mathbf{Pr}[(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) = (\boldsymbol{b}'_1, \ldots, \boldsymbol{b}'_m)]$$

Problem: Does not measure hardness of winning all uncorrupted instances.

Reason: If ∃ adversary with

$$\mathbf{Pr}[\boldsymbol{b}_1 = \boldsymbol{b}'_1] > 3/4$$

Then ∃ adversary guessing second bit at random, with

$$\mathbf{Pr}[(\boldsymbol{b}_1, \boldsymbol{b}_2) = (\boldsymbol{b}'_1, \boldsymbol{b}'_2)] > 3/4 \times 1/2 = 3/8$$

Defining mi security for encryption

Attempt #1: **AND-advantage**

LORA-security: **Output:** $(b'_1, \dots, b'_m)$

Advantage:

$$\mathbf{Adv}^{\mathrm{m-lora}}(A) = \mathbf{Pr}[(b_1, \dots, b_m) = (b'_1, \dots, b'_m)]$$

Problem: Does not measure hardness of winning all uncorrupted instances.

Reason: If ∃ adversary with

$$\mathbf{Pr}[b_1 = b'_1] > 3/4$$

Then ∃ adversary guessing second bit at random, with

$$\mathbf{Pr}[(b_1, b_2) = (b'_1, b'_2)] > 3/4 \times 1/2 = 3/8$$

Defining mi security for encryption

Attempt #2: **XOR-advantage**

Defining mi security for encryption

Attempt #2: XOR-advantage

LORX-security: **Output:** b'

Advantage:

$$\mathbf{Adv}^{\mathbf{m-lor}x}(A) = 2 \times \{\mathbf{Pr}[b' = b_1 \oplus \cdots \oplus b_m] - 1/2\}$$

Defining mi security for encryption

Attempt #2: **XOR-advantage**

LORX-security: **Output:** $\boldsymbol{b'}$

Advantage:

$$\mathbf{Adv}^{\mathbf{m-lor}x}(A) = 2 \times \{\mathbf{Pr}[\boldsymbol{b'} = \boldsymbol{b_1} \oplus \cdots \oplus \boldsymbol{b_m}] - 1/2\}$$

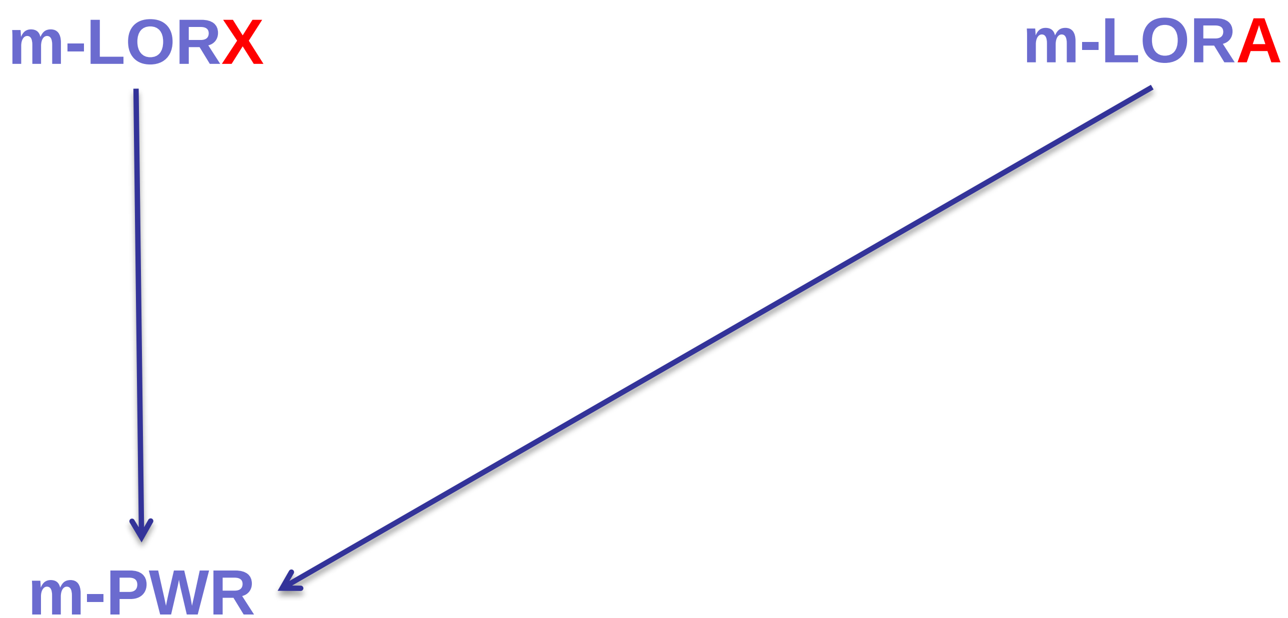
Reason: If ∃ adversary with

$$\mathbf{Pr}[\boldsymbol{b'} = \boldsymbol{b_1}] > \frac{1+\varepsilon}{2}$$

Then: Adversary guessing second bit has **no advantage**

$$\mathbf{Pr}[\boldsymbol{b'} = \boldsymbol{b_1} \oplus \boldsymbol{b_2}] = \frac{1}{2}$$

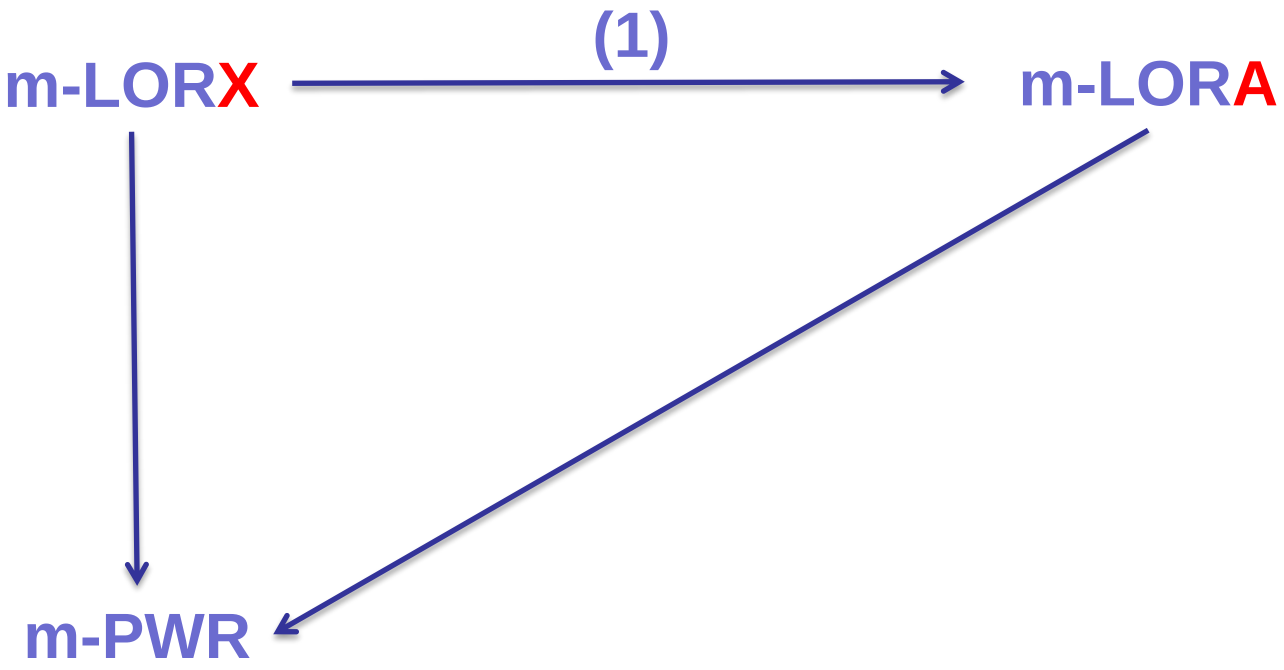
Mi security notions – Relations



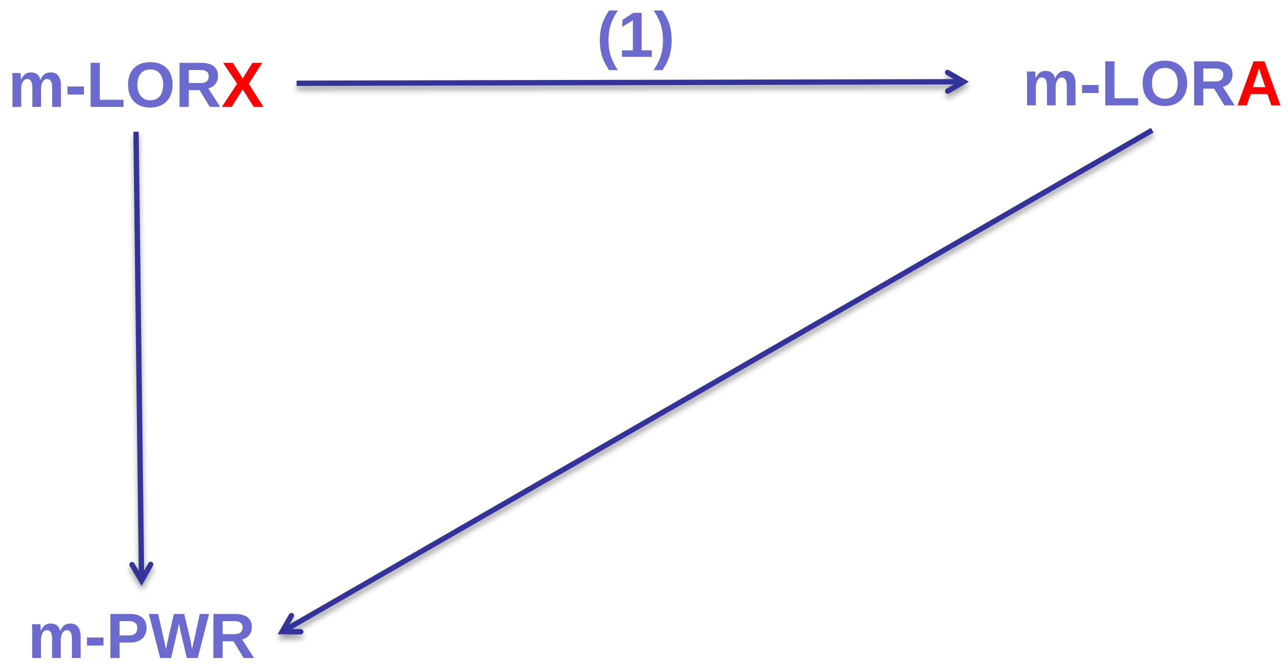
Mi security notions – Relations

$$
\begin{tikzcd}
\text{m-LORX} \arrow[r, "(1)"] \arrow[d] & \text{m-LORA} \arrow[dl] \\
\text{m-PWR} &
\end{tikzcd}
$$

Mi security notions – Relations

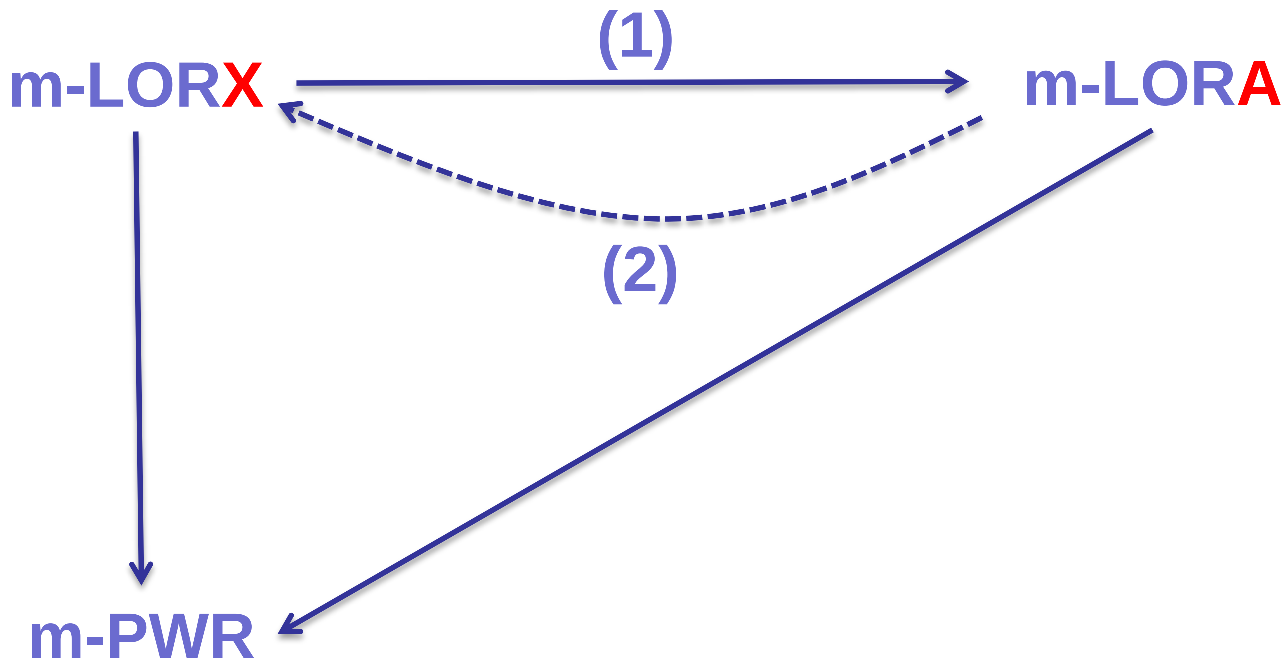


Mi security notions – Relations



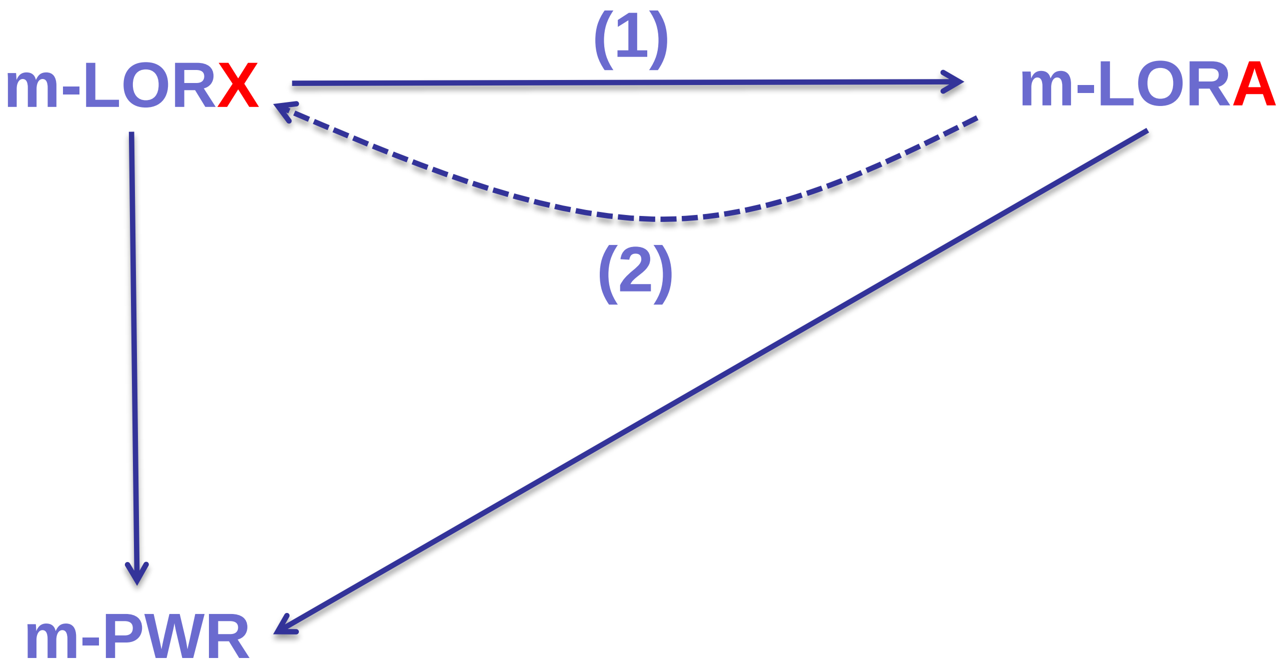
1) Holds in most cases – proof relies on probabilistic lemma from **[U09].**

Mi security notions – Relations



1) Holds in most cases – proof relies on probabilistic lemma from **[U09].**

Mi security notions – Relations

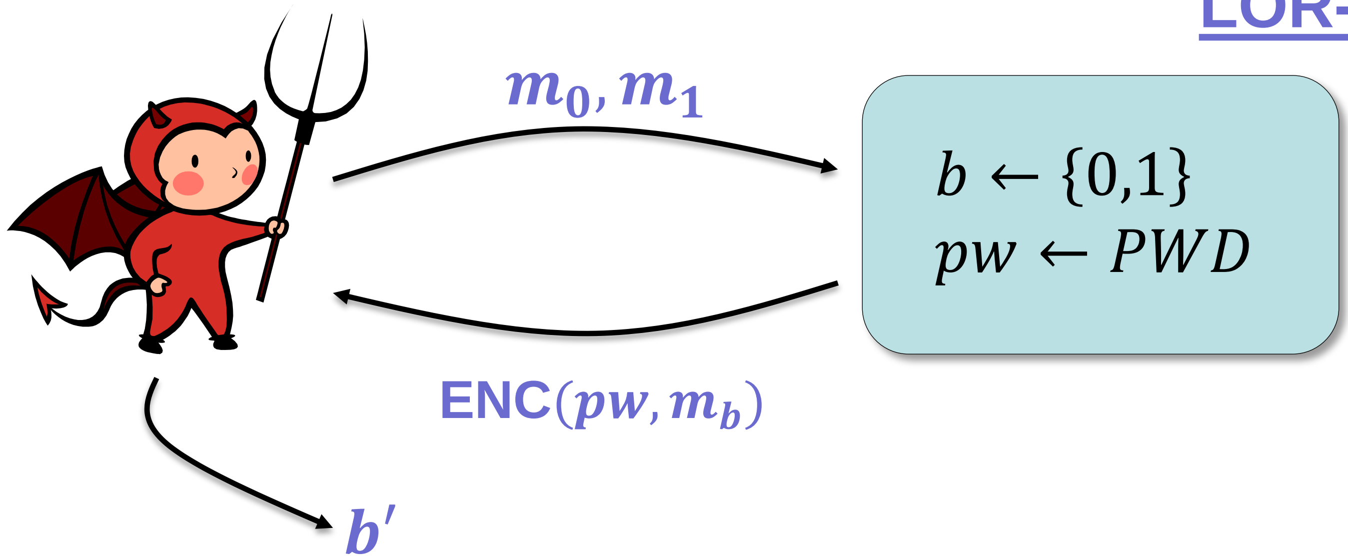


1) Holds in most cases – proof relies on probabilistic lemma from **[U09].**

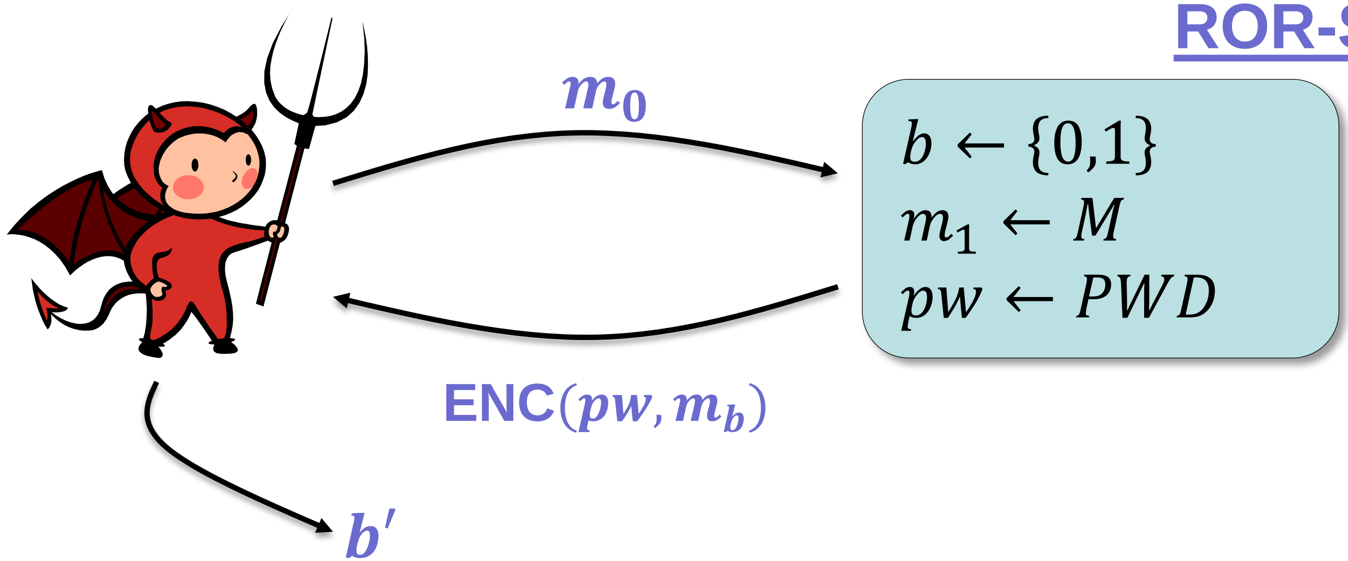
2) Very loose asymptotic implication – based on Goldreich-Levin Theorem **[GL89]**

Relations – LOR vs ROR

LOR-Security



ROR-Security



Relations – LOR vs ROR

Relations – LOR vs ROR

Classical textbook theorem.

$$\mathbf{Adv}^{\mathrm{ror}}(t) \leq \mathbf{Adv}^{\mathrm{lor}}(t) \leq 2 \times \mathbf{Adv}^{\mathrm{ror}}(t)$$

Relations – LOR vs ROR

Hybrid argument

Classical textbook theorem.

$$\mathbf{Adv}^{\text{ror}}(t) \leq \mathbf{Adv}^{\text{lor}}(t) \leq 2 \times \mathbf{Adv}^{\text{ror}}(t)$$

Relations – LOR vs ROR

Hybrid argument

Classical textbook theorem.

$$\mathbf{Adv}^{\color{red}{\mathrm{ror}}}(\boldsymbol{t}) \leq \mathbf{Adv}^{\color{red}{\mathrm{lor}}}(\boldsymbol{t}) \leq \mathbf{2} \times \mathbf{Adv}^{\color{red}{\mathrm{ror}}}(\boldsymbol{t})$$

L **R** $\leq$ **L** **\$** $+$ **\$** **R**

Relations – LOR vs ROR

Hybrid argument

Classical textbook theorem.

$$\mathbf{Adv}^{\mathrm{ror}}(t) \le \mathbf{Adv}^{\mathrm{lor}}(t) \le 2 \times \mathbf{Adv}^{\mathrm{ror}}(t)$$

L R $\le$ L \$ $+$ \$ R

Mi setting with m instances:

$$\mathbf{Adv}^{\mathrm{m-rorx}}(t) \le \mathbf{Adv}^{\mathrm{m-lorx}}(t) \le 2^{m} \times \mathbf{Adv}^{\mathrm{m-rorx}}(t)$$

Relations – LOR vs ROR

Hybrid argument

Classical textbook theorem.

$$\mathbf{Adv}^{\mathrm{ror}}(t) \le \mathbf{Adv}^{\mathrm{lor}}(t) \le 2 \times \mathbf{Adv}^{\mathrm{ror}}(t)$$

$$\boxed{L}\ \boxed{R} \quad \le \quad \boxed{L}\ \boxed{\$} \quad + \quad \boxed{\$}\ \boxed{R}$$

Mi setting with m instances:

$$\mathbf{Adv}^{\mathrm{m-rorx}}(t) \le \mathbf{Adv}^{\mathrm{m-lorx}}(t) \le 2^{m} \times \mathbf{Adv}^{\mathrm{m-rorx}}(t)$$

$$\begin{array}{cccccccccccccc} \boxed{L} & \boxed{R} & & \boxed{L} & \boxed{\$} & & \boxed{\$} & \boxed{R} & & \boxed{\$} & \boxed{R} & & \boxed{L} & \boxed{\$} \\ & & \le & & & + & & & + & & & + & & \\ \boxed{L} & \boxed{R} & & \boxed{L} & \boxed{\$} & & \boxed{L} & \boxed{\$} & & \boxed{\$} & \boxed{R} & & \boxed{\$} & \boxed{R} \end{array}$$

Relations – LOR vs ROR

Hybrid argument

Classical textbook theorem.

$$\mathbf{Adv}^{\mathrm{ror}}(t) \leq \mathbf{Adv}^{\mathrm{lor}}(t) \leq 2 \times \mathbf{Adv}^{\mathrm{ror}}(t)$$

Tight!

Mi setting with m instances:

$$\mathbf{Adv}^{\mathrm{m-rorx}}(t) \leq \mathbf{Adv}^{\mathrm{m-lorx}}(t) \leq 2^{m} \times \mathbf{Adv}^{\mathrm{m-rorx}}(t)$$

Outline

1. Multi-instance security

2. Security of PKCS#5 – A case study

Outline

1. Multi-instance security

2. Security of PKCS#5 – A case study

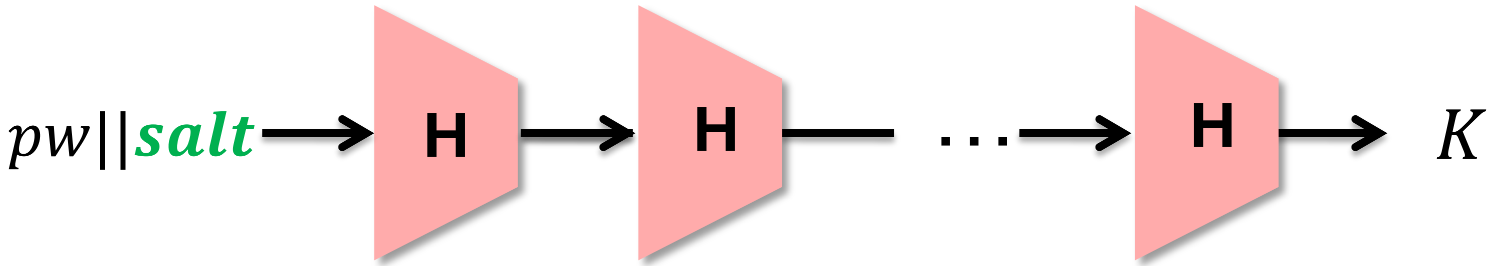
PKCS#5 – Defining KDF Security

PKCS#5 – Defining KDF Security

Question: Does salting provably ensures multi-instance security amplification? **YES!**

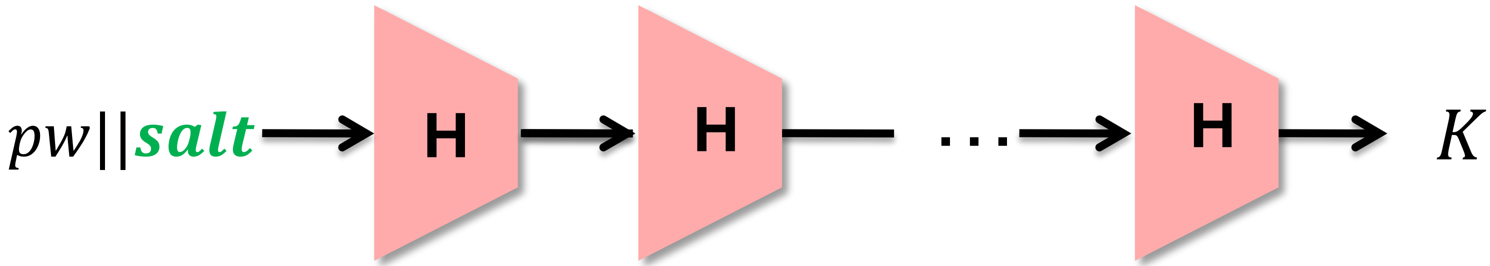
PKCS#5 – Defining KDF Security

Question: Does salting provably ensures multi-instance security amplification? **YES!**



PKCS#5 – Defining KDF Security

Question: Does salting provably ensures multi-instance security amplification? **YES!**

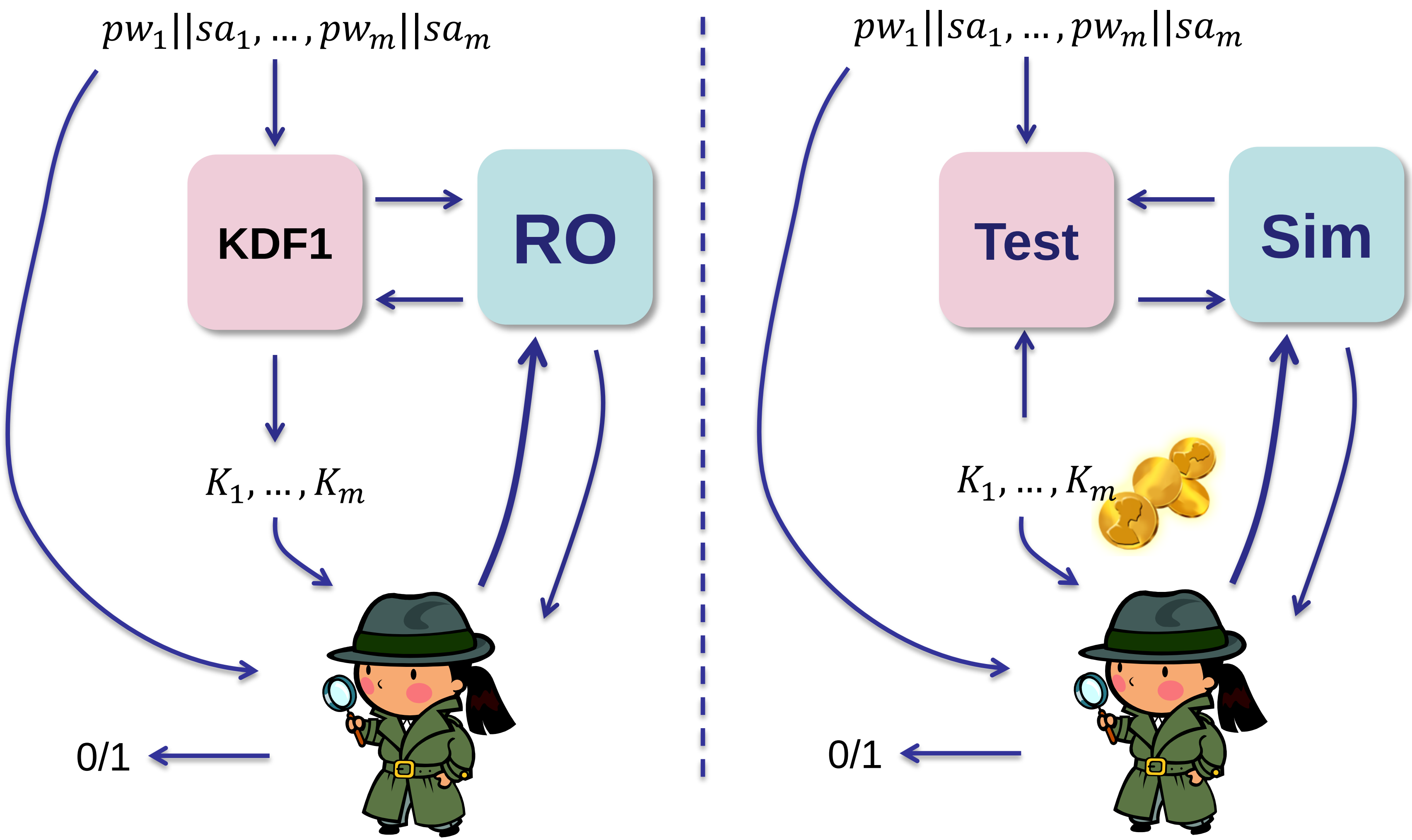


Main step:
Security analysis of **KDF1** for case **H** = **RO**.

KDF Security in the ROM

KDF satisfies indifferentiability-like poperty [MRH04]

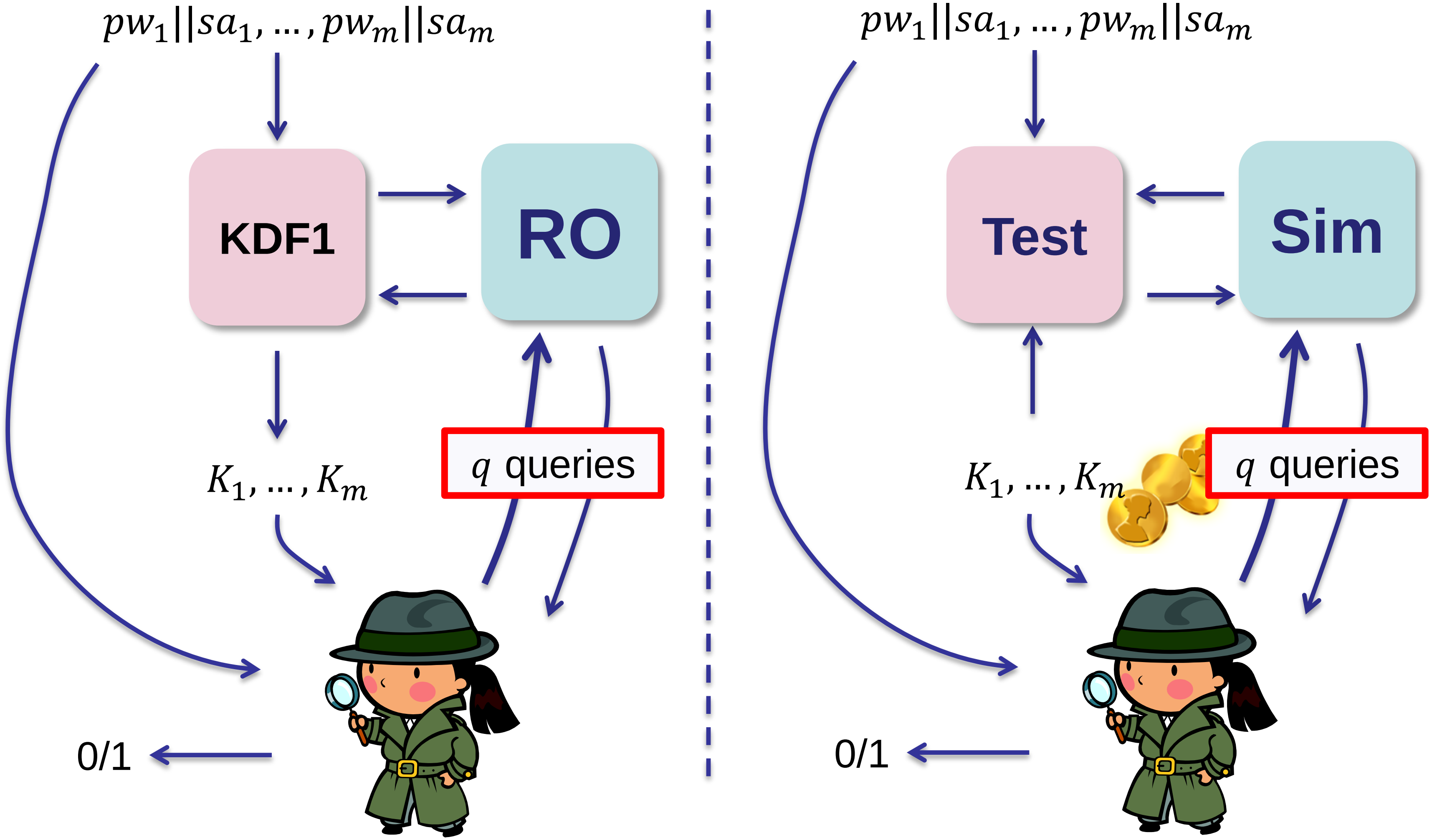
∃Sim ∀ password distributions: Left ≈ Right



KDF Security in the ROM

KDF satisfies indifferentiability-like poperty [MRH04]

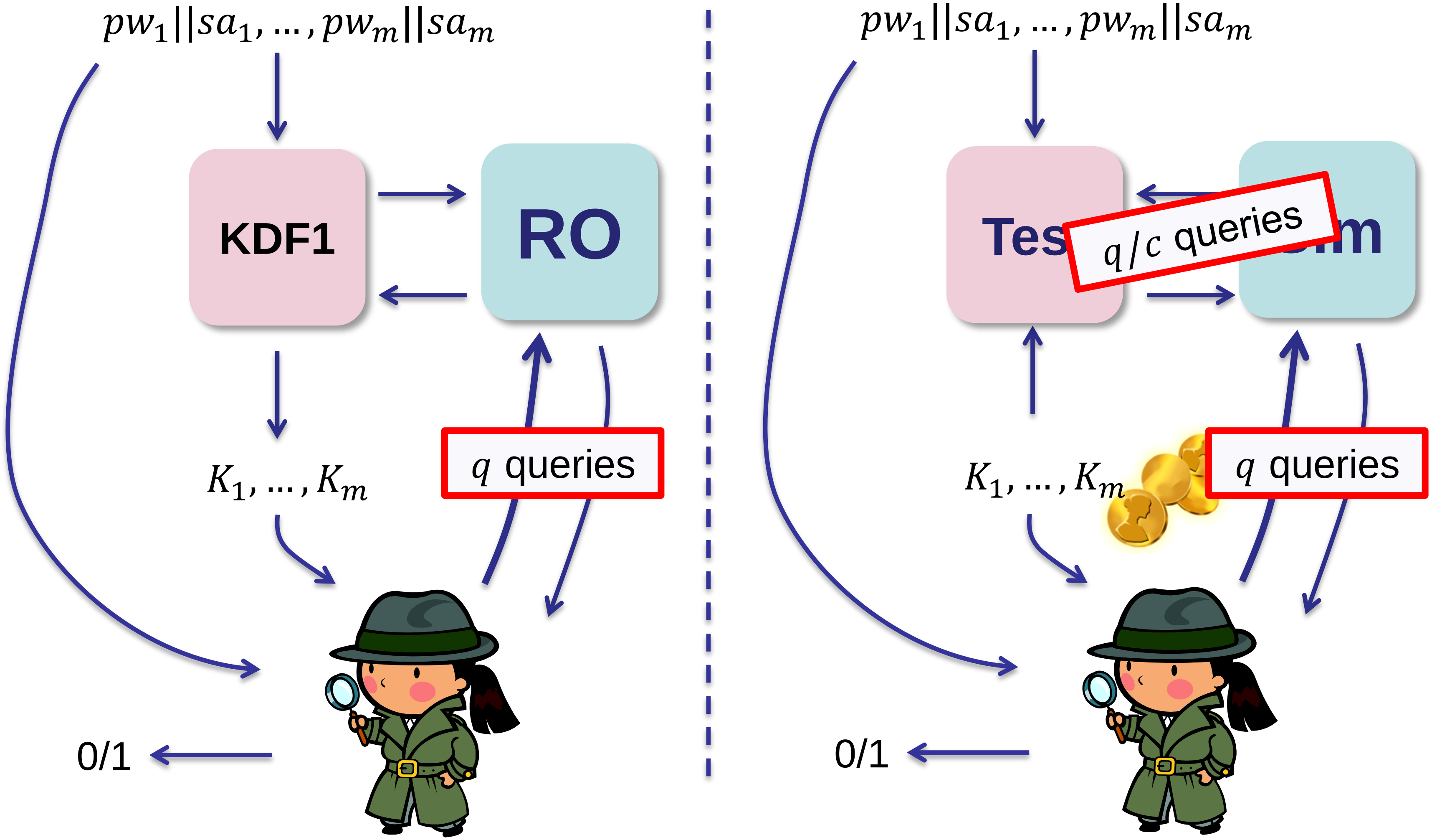
$\exists$Sim $\forall$ password distributions: Left $\approx$ Right



KDF Security in the ROM

KDF satisfies indifferentiability-like poperty [MRH04]

$\exists$Sim $\forall$ password distributions: Left $\approx$ Right



Final result: Security of PB-Encrypt

Question: Does salting deliver multi-instance security amplification for PKCS#5?

PB-Encrypt(pw, M)
$salt \leftarrow \{0,1\}^s$
$K \leftarrow$ **H**$^c(pw||salt)$
$C \leftarrow$ **ENC**(K, M)
Return $C||salt$

Theorem: ∀A making q RO queries, ∃ B such that

$$\mathbf{Adv}^{\mathbf{m-rorx}}_{\text{PB-Encrypt}}(A) < \frac{q}{mcN} + m \cdot \mathbf{Adv}^{\mathbf{ror}}_{\mathbf{ENC}}(B) + \frac{q^2}{2^n} + \frac{q^2}{2^s}$$

Final result: Security of PB-Encrypt

Question: Does salting deliver multi-instance security amplification for PKCS#5?

PB-Encrypt(pw, M)
$salt \leftarrow \{0,1\}^s$
$K \leftarrow \mathbf{H}^c(pw||salt)$
$C \leftarrow \mathbf{ENC}(K, M)$
Return $C||salt$

Theorem: ∀A making q RO queries, ∃ B such that

$$\mathbf{Adv}^{\mathbf{m-rorx}}_{\mathbf{PB-Encrypt}}(A) < \frac{q}{mcN} + m \cdot \mathbf{Adv}^{\mathbf{ror}}_{\mathbf{ENC}}(B) + \frac{q^2}{2^n} + \frac{q^2}{2^s}$$

Work $m \times c \times N$ to break encryption (RO queries)

Concluding Remarks

Summary:

Concluding Remarks

Summary:

- The world has **multiple users**

Concluding Remarks

Summary:

- The world has **multiple users**
- **Weak individual instances** sometimes unavoidable

Concluding Remarks

Summary:

- The world has **multiple users**
- **Weak individual instances** sometimes unavoidable
- Mi security as a second line of defense

Concluding Remarks

Summary:

- The world has **multiple users**
- **Weak individual instances** sometimes unavoidable
- Mi security as a second line of defense
- Interesting technical questions

Concluding Remarks

Summary:

- The world has **multiple users**
- **Weak individual instances** sometimes unavoidable
- Mi security as a second line of defense
- Interesting technical questions
- **First security analysis** of PKCS#5 in the mi setting

Concluding Remarks

Summary:

- The world has **multiple users**
- **Weak individual instances** sometimes unavoidable
- Mi security as a second line of defense
- Interesting technical questions
- **First security analysis** of PKCS#5 in the mi setting

Thank you!