Count Corruptions, Not Users:

Improved Tightness for Signatures, Encryption and Authenticated Key Exchange

Mihir Bellare, Doreen Riepel, Stefano Tessaro, Yizhao Zhang

ASIACRYPT 2024

Motivation

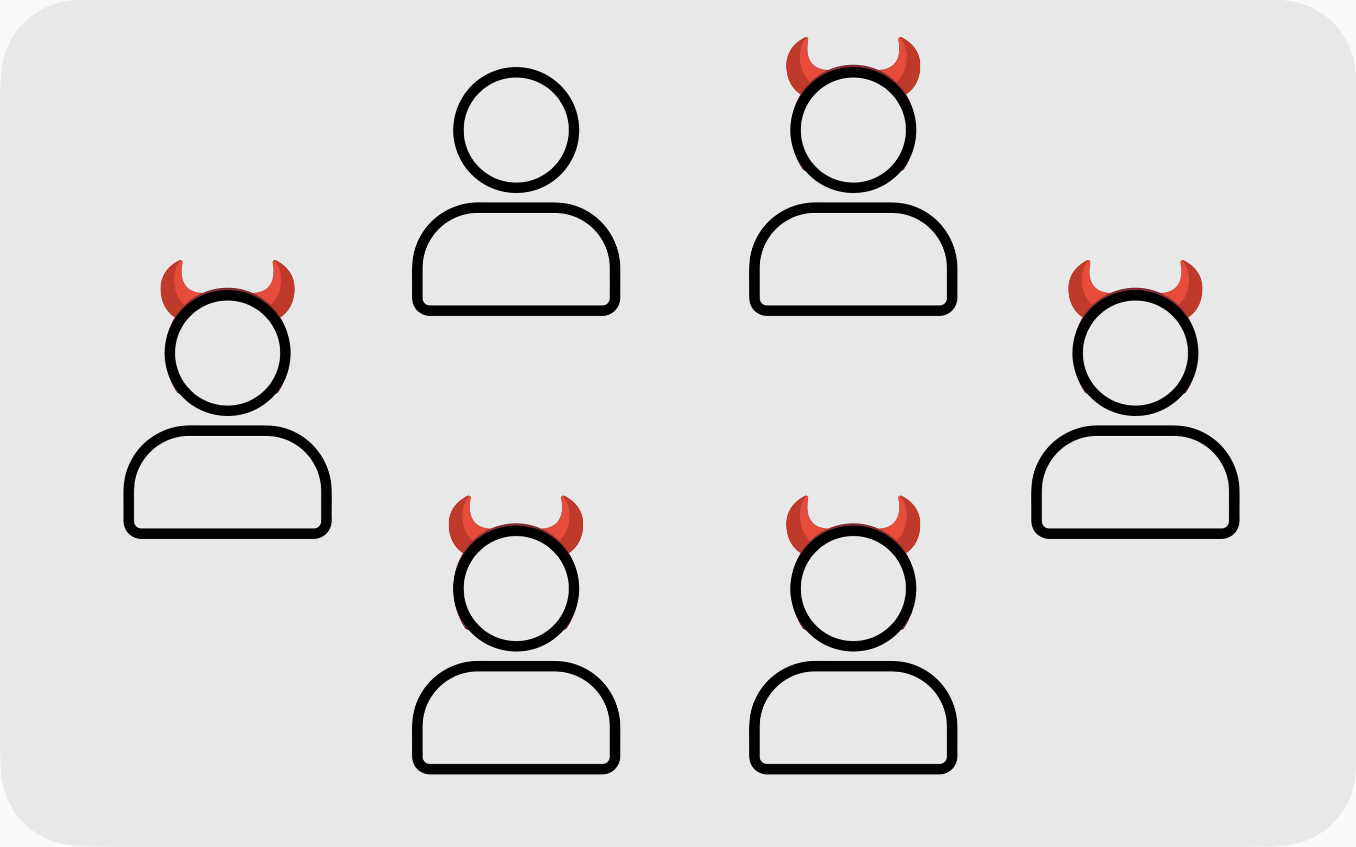
Modern applications ask for security in the presence of powerful adversaries who may **adaptively corrupt** parties.

- Key exchange (TLS), messaging (Signal, MLS), etc.

Motivation

Modern applications ask for security in the presence of powerful adversaries who may **adaptively corrupt** parties.

- Key exchange (TLS), messaging (Signal, MLS), etc.



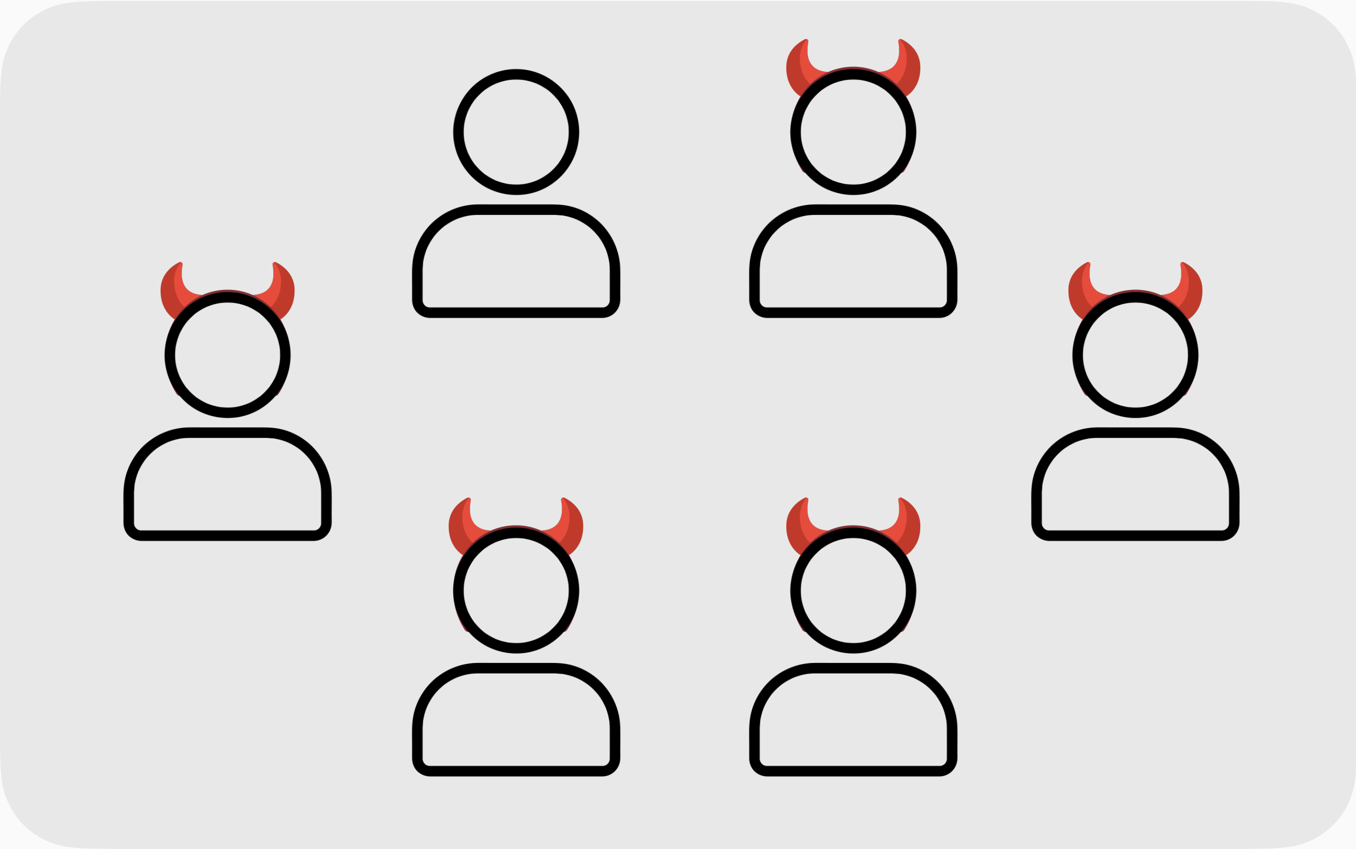
Typical model: muc security (multi-user with corruptions)

- n users
- $n - 1$ corruptions

Motivation

Modern applications ask for security in the presence of powerful adversaries who may **adaptively corrupt** parties.

- Key exchange (TLS), messaging (Signal, MLS), etc.



Typical model: muc security (multi-user with corruptions)

- n users
- $n-1$ corruptions

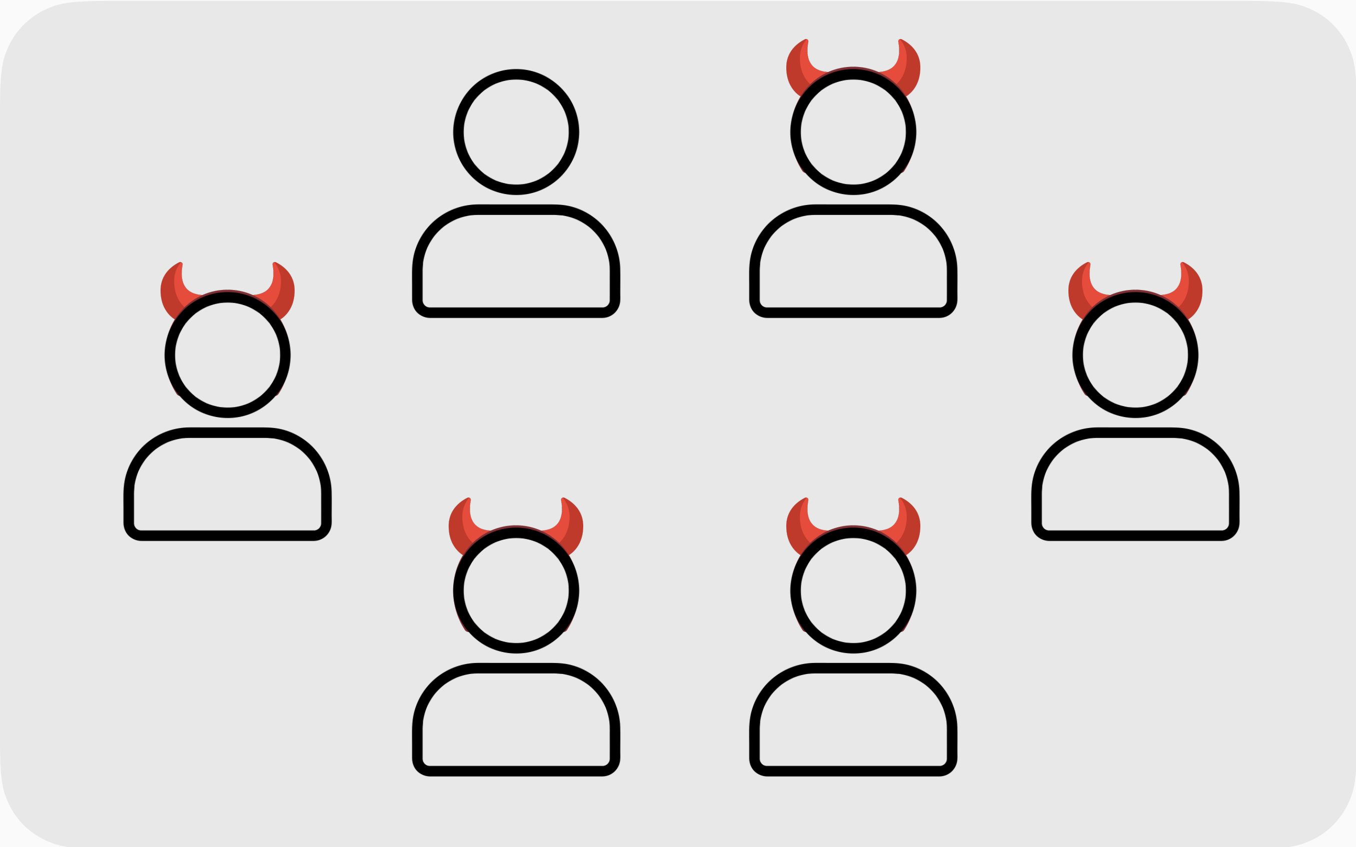
Corruptions happen, but **the number is likely small:**

- Key-owners have high incentive to prevent exposure and take significant steps
- Internet services are increasingly storing their TLS signing keys in hardware security modules
- Use of threshold cryptography

Motivation

Modern applications ask for security in the presence of powerful adversaries who may **adaptively corrupt** parties.

- Key exchange (TLS), messaging (Signal, MLS), etc.



Typical model: muc security (multi-user with corruptions)

- n users
- $n-1$ corruptions

Corruptions happen, but **the number is likely small:**

- Key-owners have high incentive to prevent exposure and take significant steps
- Internet services are increasingly storing their TLS signing keys in hardware security modules
- Use of threshold cryptography

Microsoft Storm-0885 attack (2023)[1]

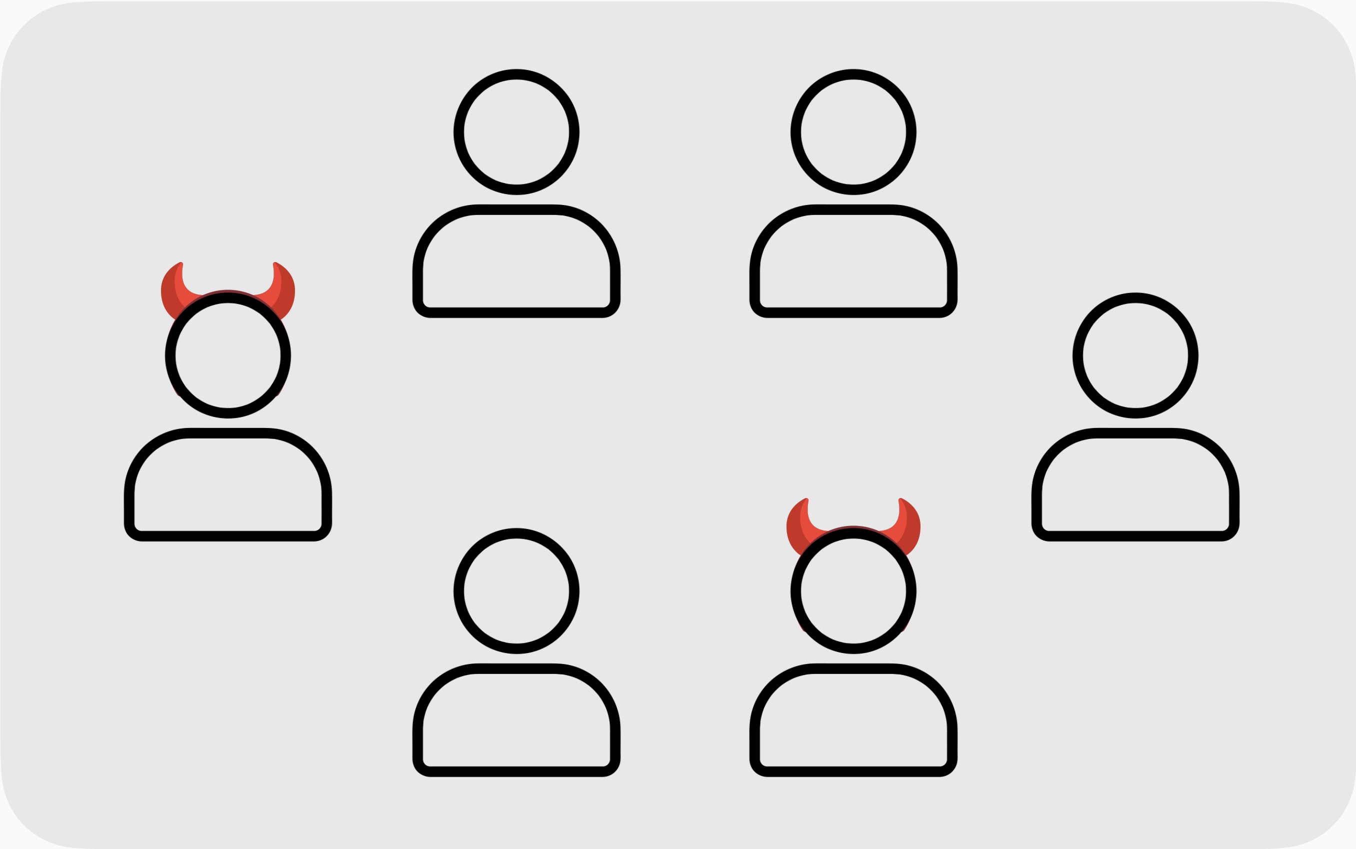
- Attackers acquired a Microsoft account (MSA) consumer signing key used to authenticate tokens
- Affected were email accounts of 22 organizations and 500 individuals globally (e.g. top-tier US government officials)

[1] https://www.microsoft.com/en-us/security/blog/2023/07/14/analysis-of-storm-0558-techniques-for-unauthorized-email-access/

Motivation

Our model: cp-muc security ("corruption-parametrized")

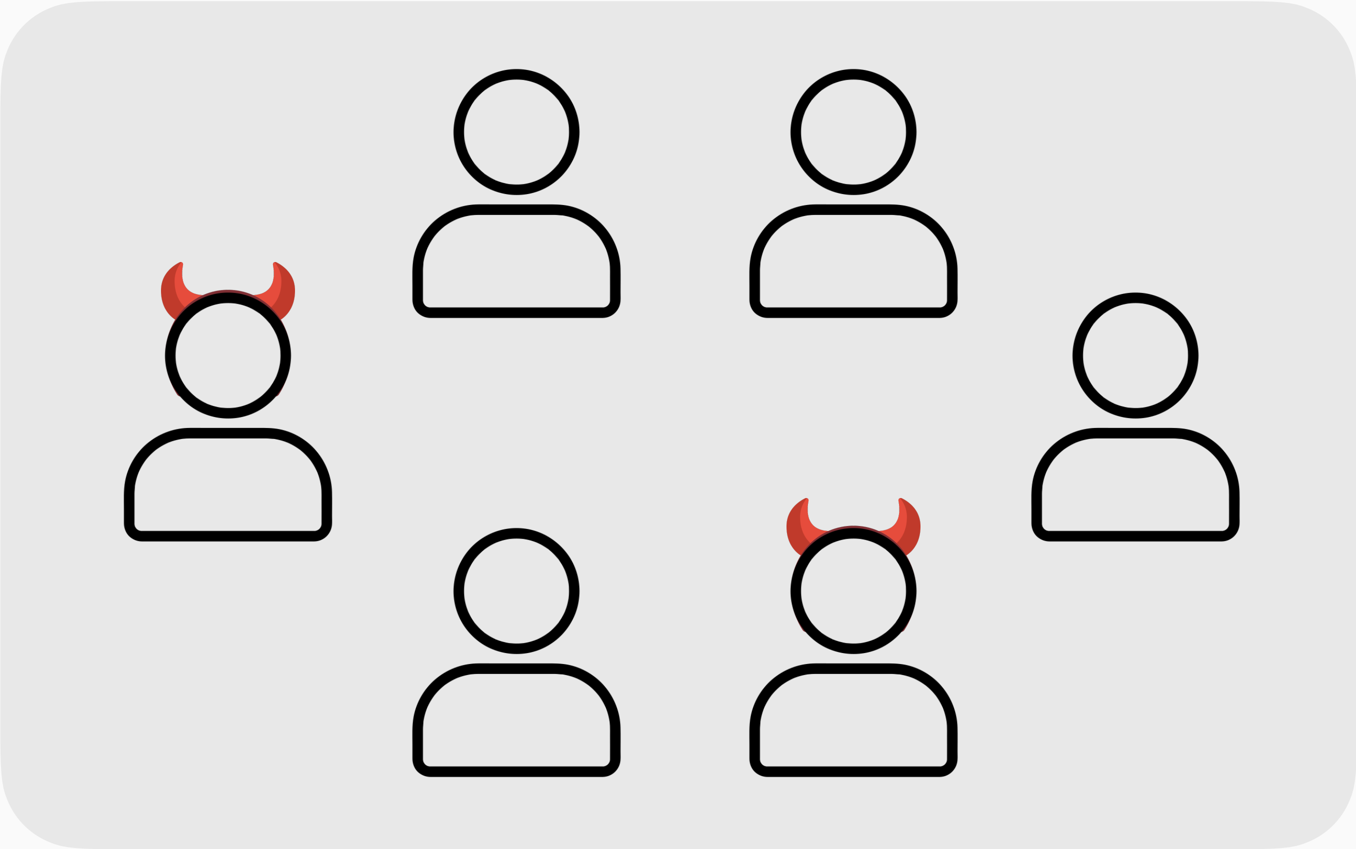
- n users
- c corruptions for $c \ll n$



Motivation

Our model: cp-muc security ("corruption-parametrized")

- n users
- c corruptions for $c \ll n$



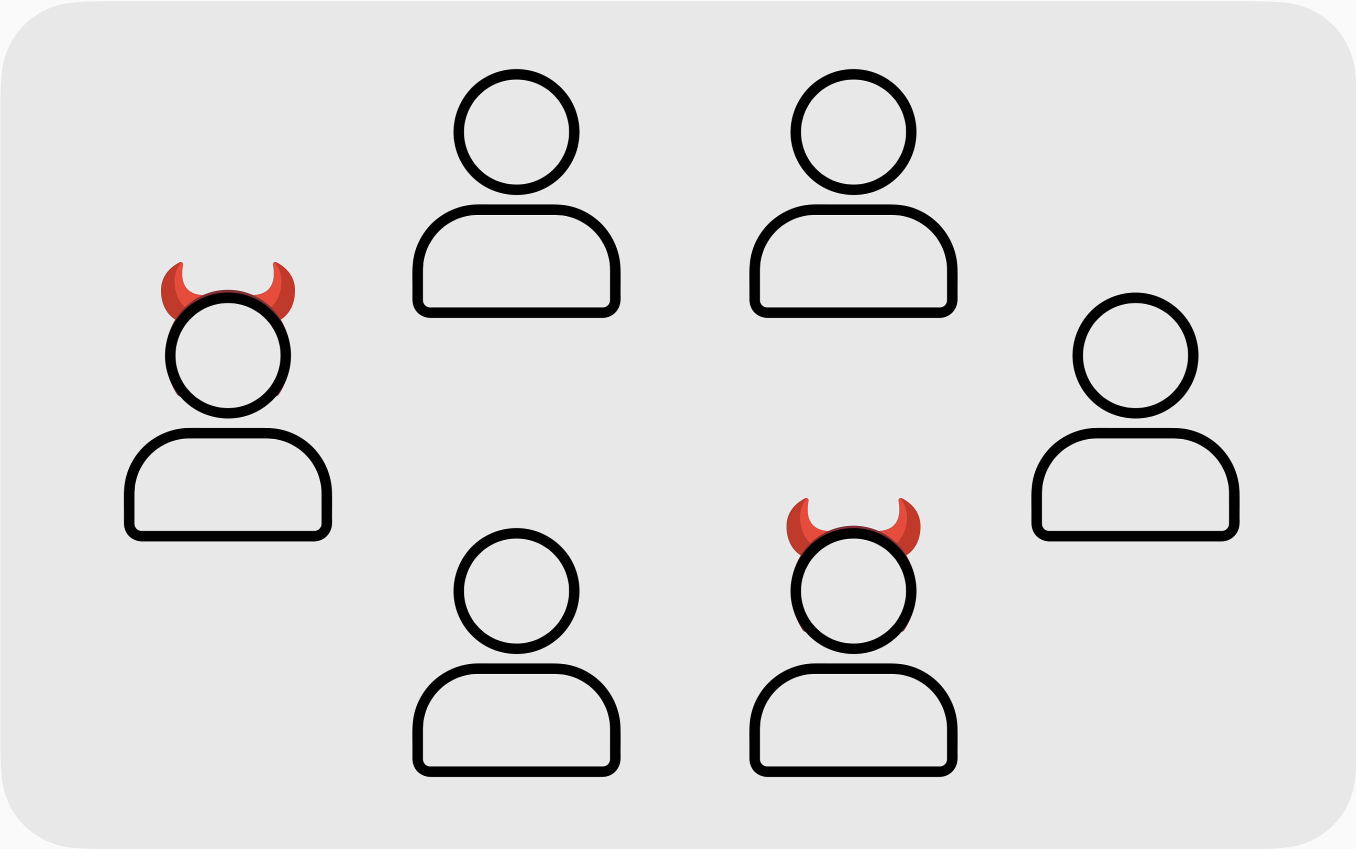
Applications

- Signing, public-key and secret-key encryption, key exchange, ...
- Similar to a "threshold" in secret sharing of MPC

Motivation

Our model: cp-muc security ("corruption-parametrized")

- n users
- c corruptions for $c \ll n$



Applications

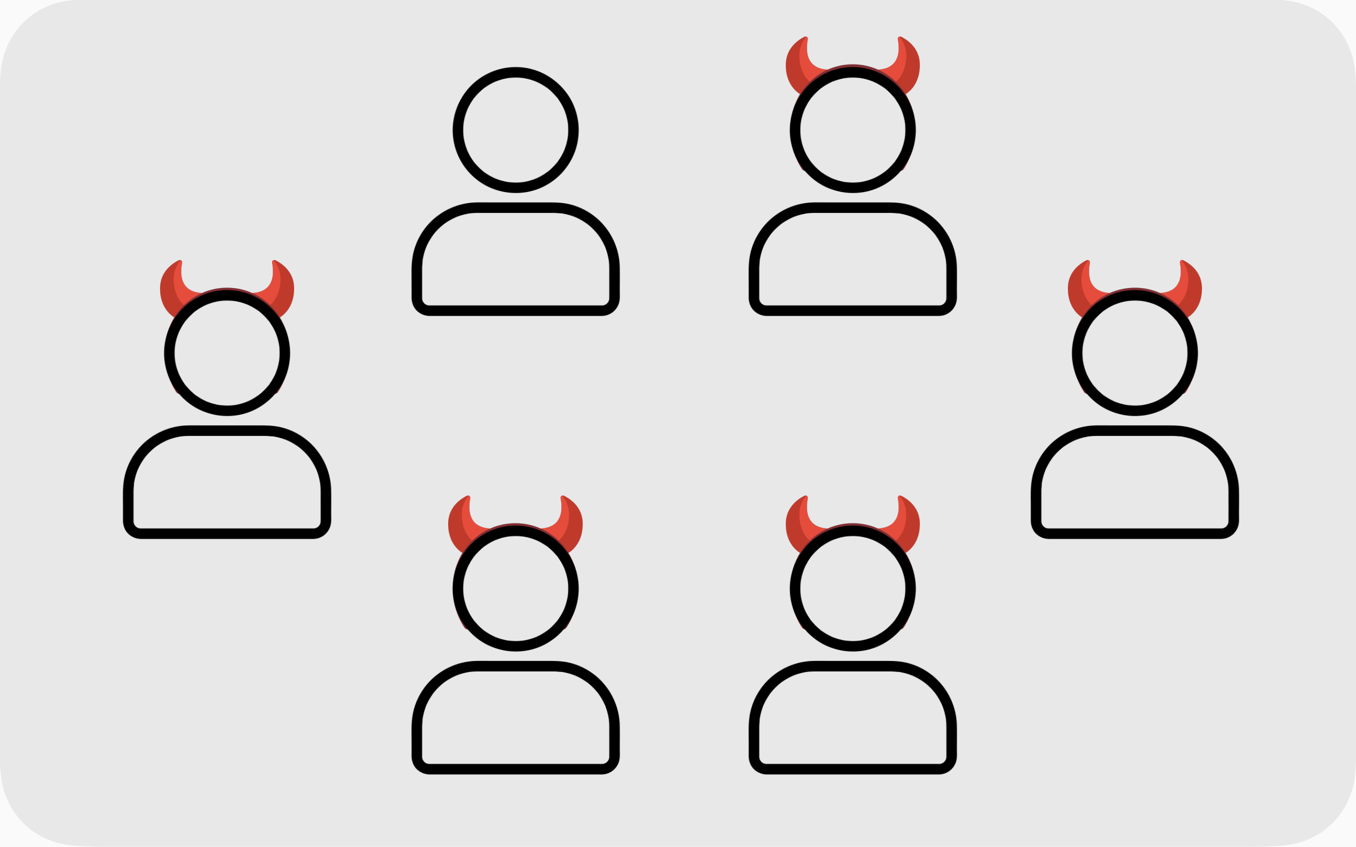
- Signing, public-key and secret-key encryption, key exchange, ...
- Similar to a "threshold" in secret sharing of MPC

Goal

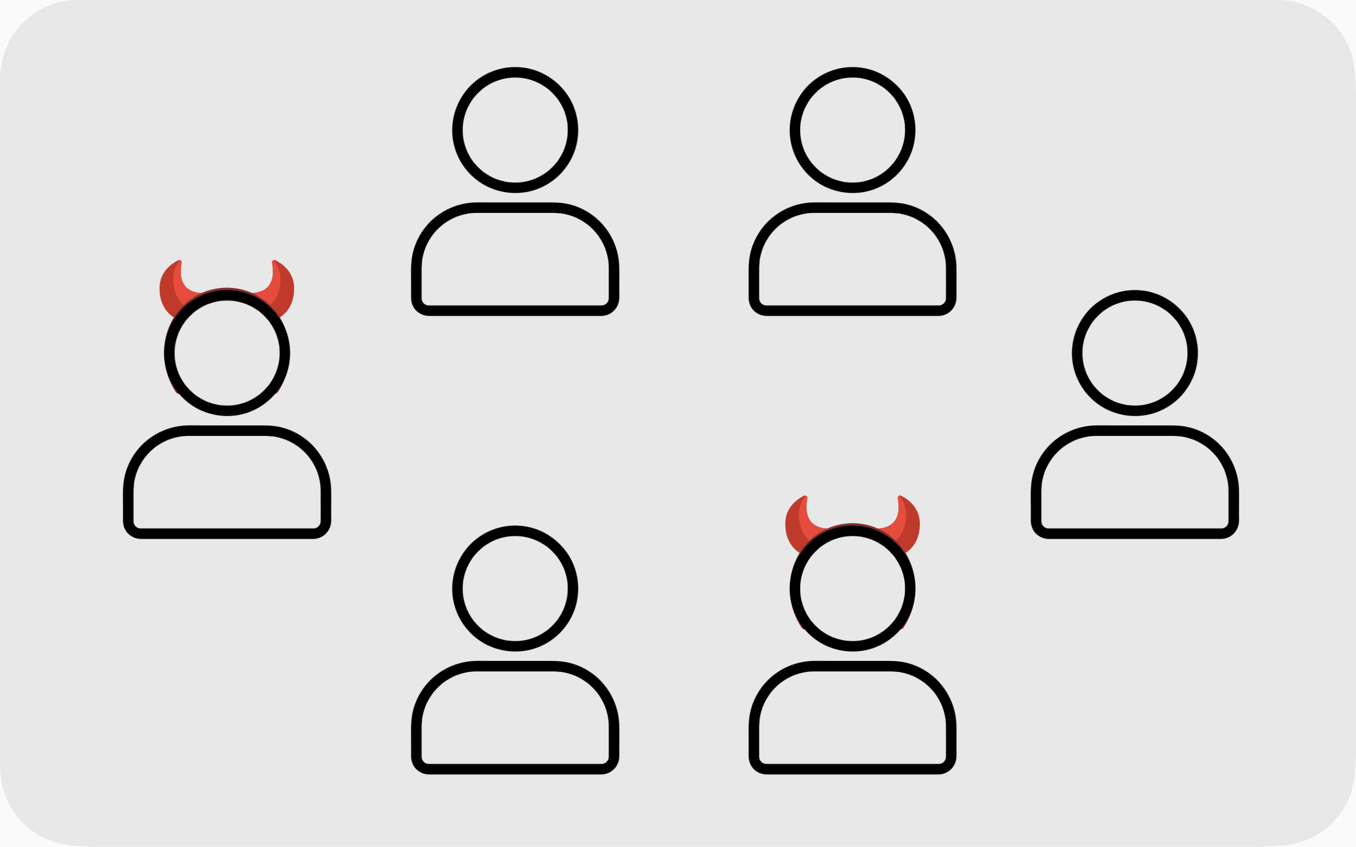
- Better concrete security guarantees for protocols deployed in practice, where otherwise tight(er) bounds are unknown or impossible

Motivation

muc security

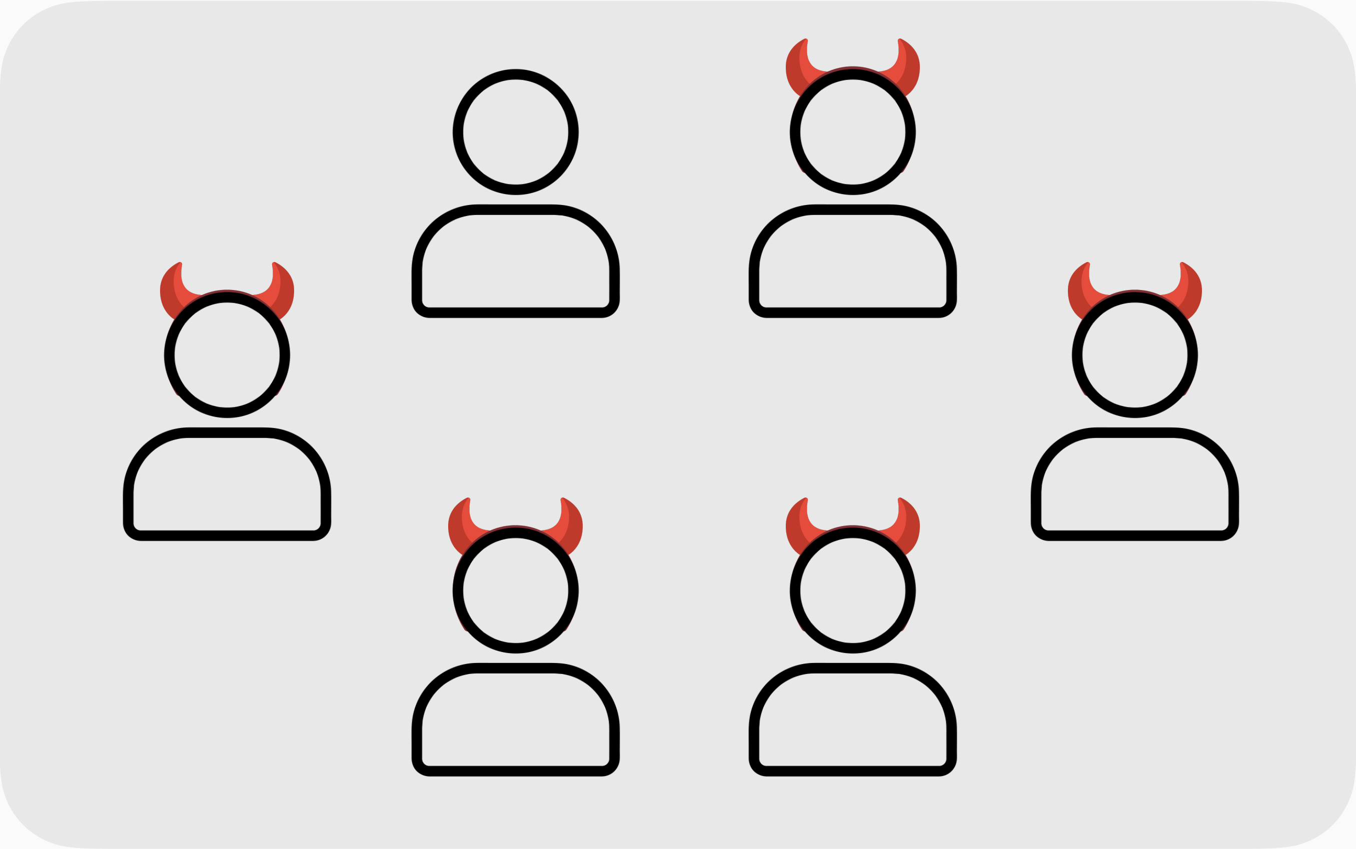


cp-muc security

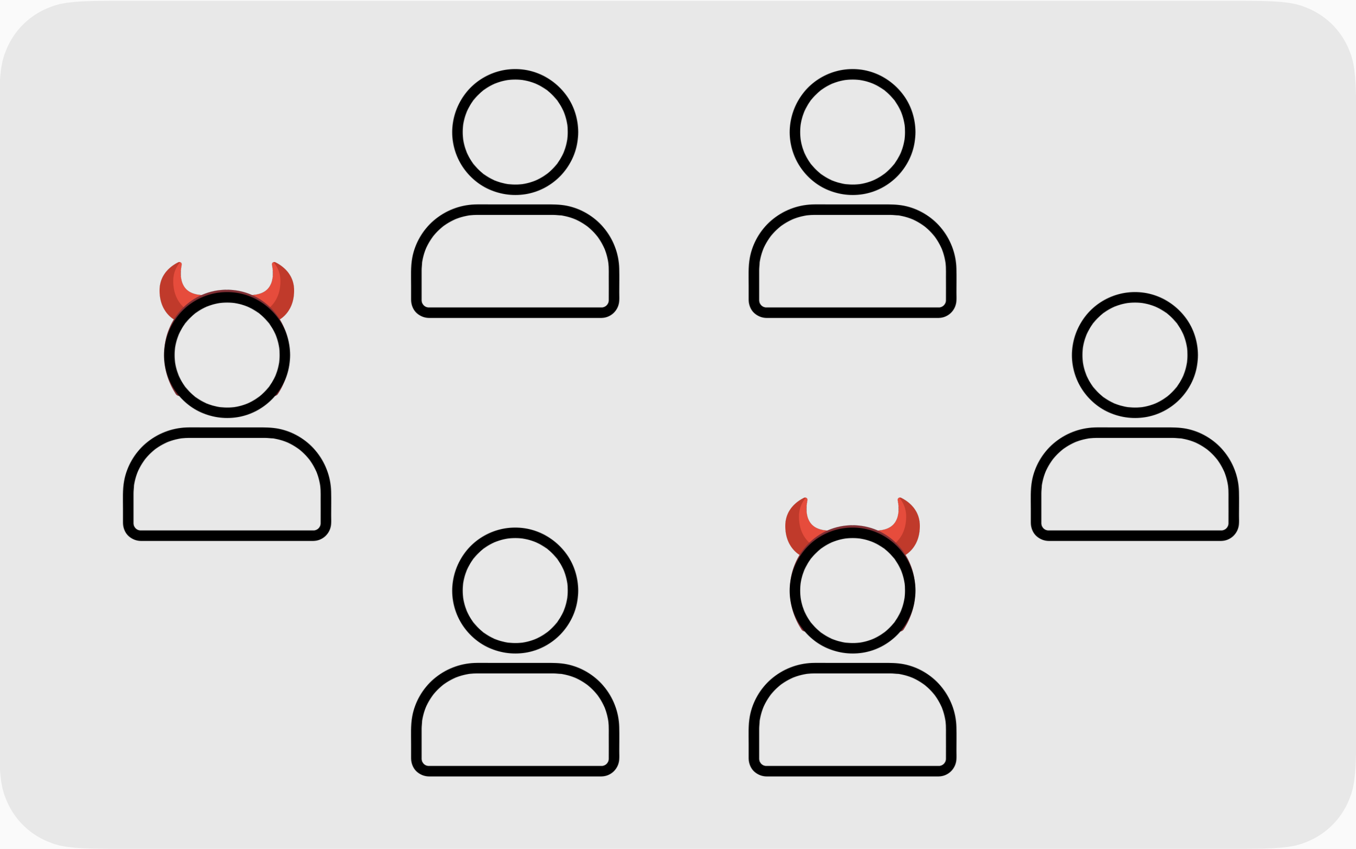


Motivation

muc security



cp-muc security

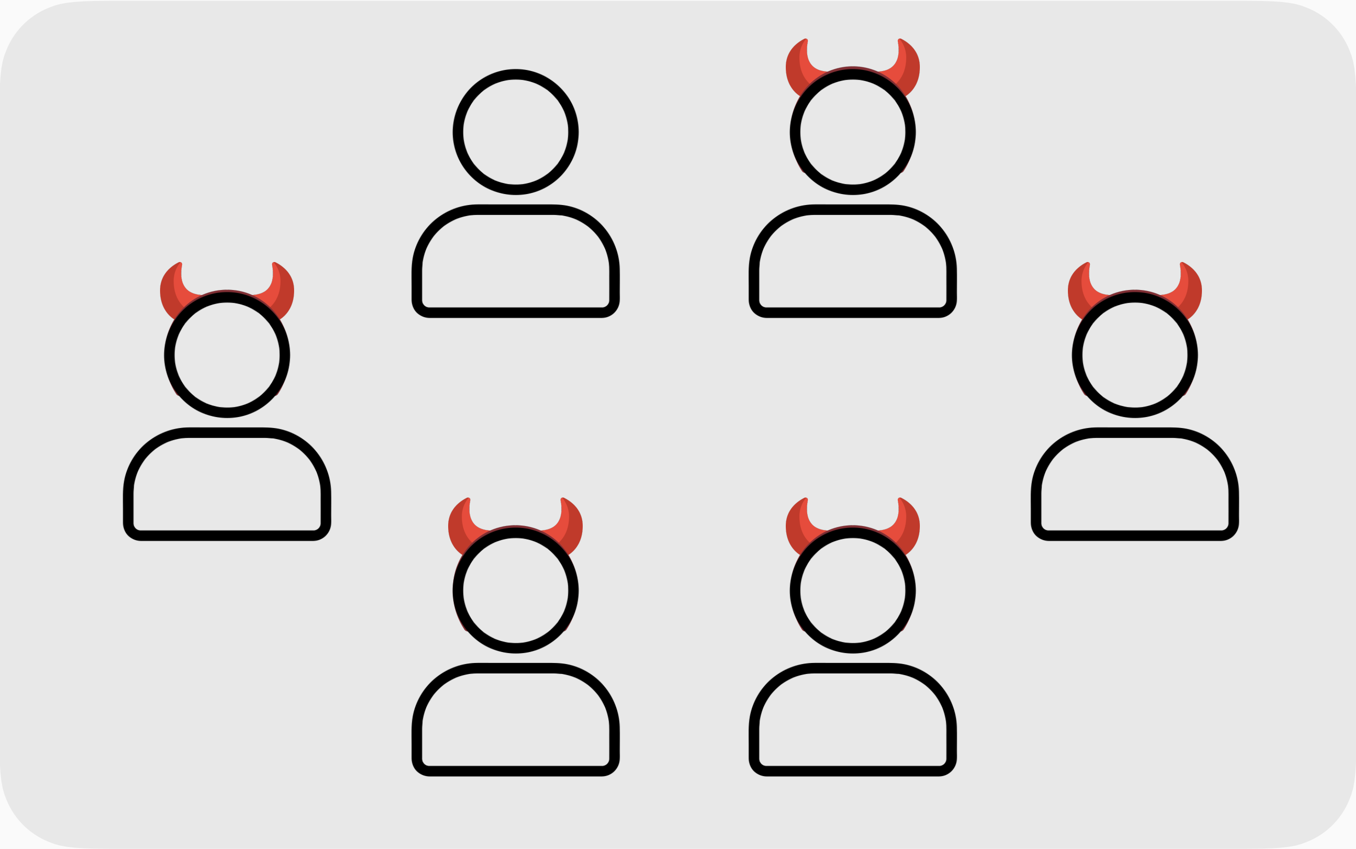


Standard hybrid argument:

- Reduces to single-user (su) security
- Security loss linear in the number of users

Motivation

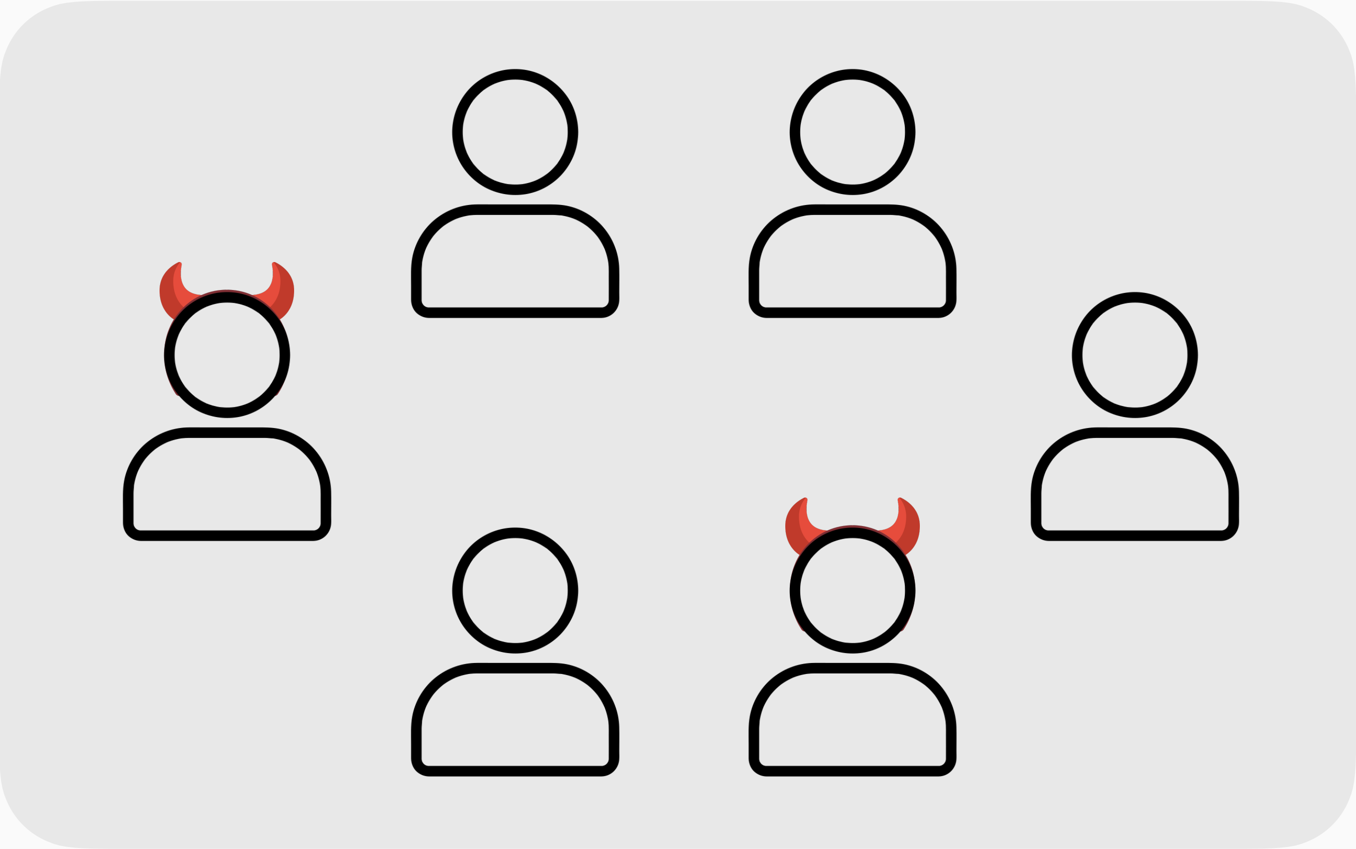
muc security



Standard hybrid argument:

- Reduces to single-user (su) security
- Security loss linear in the number of users

cp-muc security

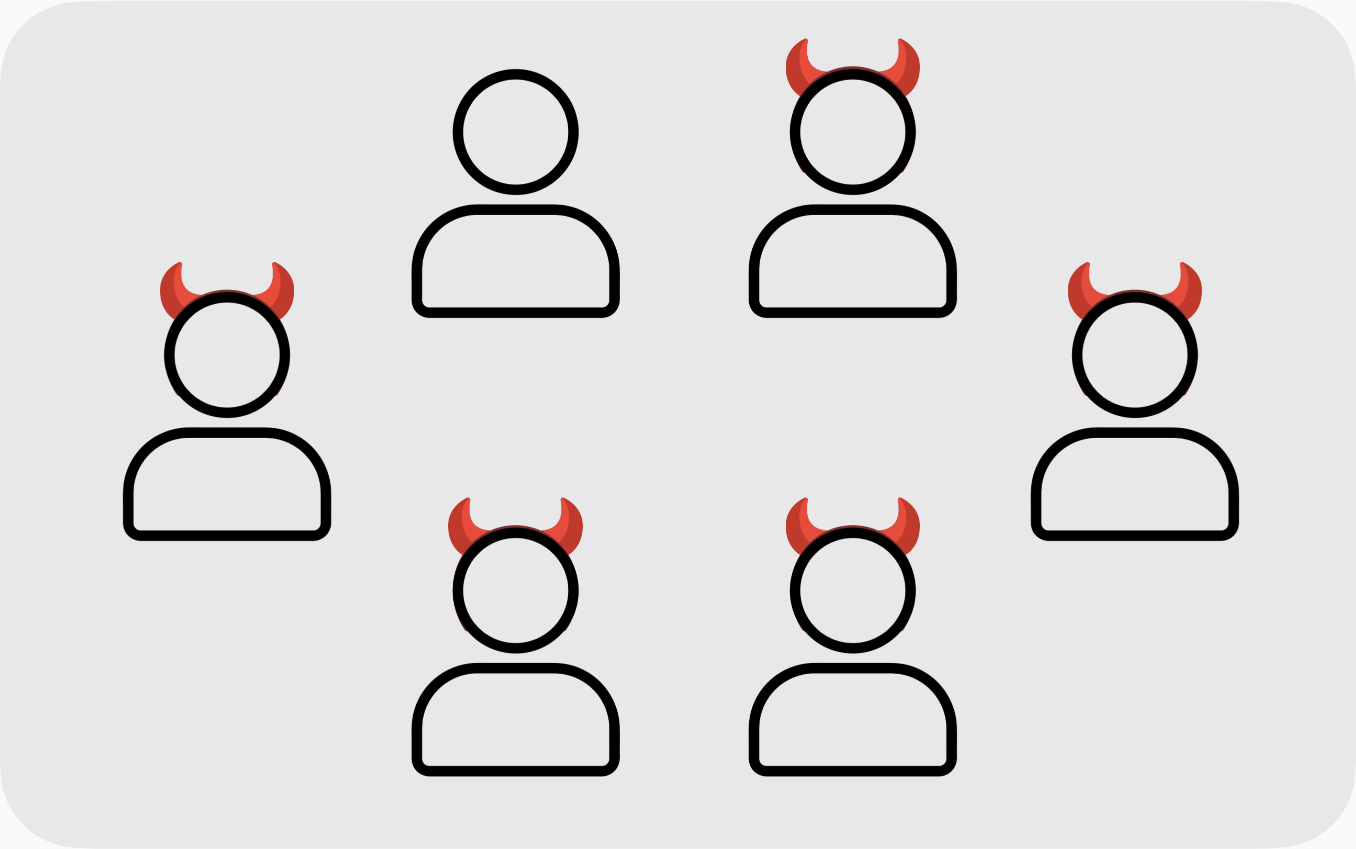


Our hope:

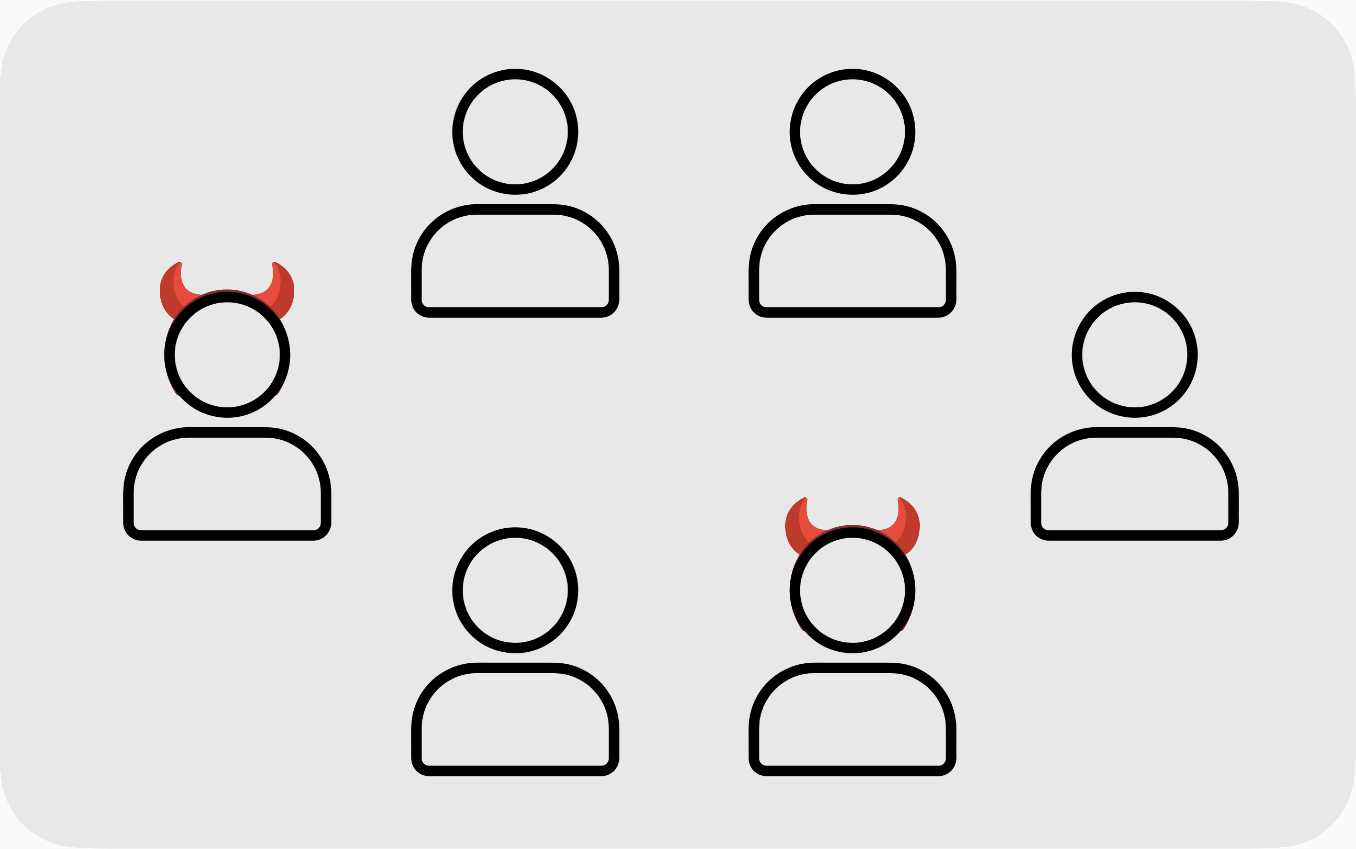
- Security loss linear in the number of corruptions

Motivation

muc security



cp-muc security



Standard hybrid argument:

- Reduces to single-user (su) security
- Security loss linear in the number of users

Our hope:

- Security loss linear in the number of corruptions

Main question:

Can we give a general theorem? Under which conditions?

Overview of our Results

Formal security specifications

- Syntax that translates into a single-user (su), multi-user (mu) and corruptions (muc) game

Overview of our Results

Formal security specifications

- Syntax that translates into a single-user (su), multi-user (mu) and corruptions (muc) game

Hamming-weight determined samplers

- Technical tool that we introduce
- Essentially it determines how a (suitable) subset of users is picked

Overview of our Results

Formal security specifications

- Syntax that translates into a single-user (su), multi-user (mu) and corruptions (muc) game

Hamming-weight determined samplers

- Technical tool that we introduce
- Essentially it determines how a (suitable) subset of users is picked

General cp-muc theorem (applies to all games which satisfy “locality” property)

- Basically all one-way (OW) games
- Indistinguishability (IND) games with independent challenge bits

Overview of our Results

Formal security specifications

- Syntax that translates into a single-user (su), multi-user (mu) and corruptions (muc) game

Hamming-weight determined samplers

- Technical tool that we introduce
- Essentially it determines how a (suitable) subset of users is picked

General cp-muc theorem (applies to all games which satisfy "locality" property)

- Basically all one-way (OW) games
- Indistinguishability (IND) games with independent challenge bits

Indirect applications of the cp-muc theorem (specialized results for "non-local" and "more advanced" games)

- IND-CCA with a single challenge bit across users (via FO, Hashed ElGamal)
- AKE protocols
- Selective opening security

Overview of our Results

Formal security specifications

- Syntax that translates into a single-user (su), multi-user (mu) and corruptions (muc) game

Hamming-weight determined samplers

- Technical tool that we introduce
- Essentially it determines how a (suitable) subset of users is picked

General cp-muc theorem (applies to all games which satisfy "locality" property)

- Basically all one-way (OW) games
- Indistinguishability (IND) games with independent challenge bits

Indirect applications of the cp-muc theorem (specialized results for "non-local" and "more advanced" games)

- IND-CCA with a single challenge bit across users (via FO, Hashed ElGamal)
- AKE protocols
- Selective opening security

We also give matching optimality (impossibility) results for a large class of games and schemes.

Overview of our Results

Formal security specifications

- Syntax that translates into a single-user (su), multi-user (mu) and corruptions (muc) game

Hamming-weight determined samplers

- Technical tool that we introduce
- Essentially it determines how a (suitable) subset of users is picked

General cp-muc theorem (applies to all games which satisfy "locality" property)

- Basically all one-way (OW) games
- Indistinguishability (IND) games with independent challenge bits

Main focus of this talk (using the example of UF-CMA secure signatures)

Indirect applications of the cp-muc theorem (specialized results for "non-local" and "more advanced" games)

- IND-CCA with a single challenge bit across users (via FO, Hashed ElGamal)
- AKE protocols
- Selective opening security

We also give matching optimality (impossibility) results for a large class of games and schemes.

Digital Signatures

Syntax: A signature scheme Sig is described via algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$.

Digital Signatures

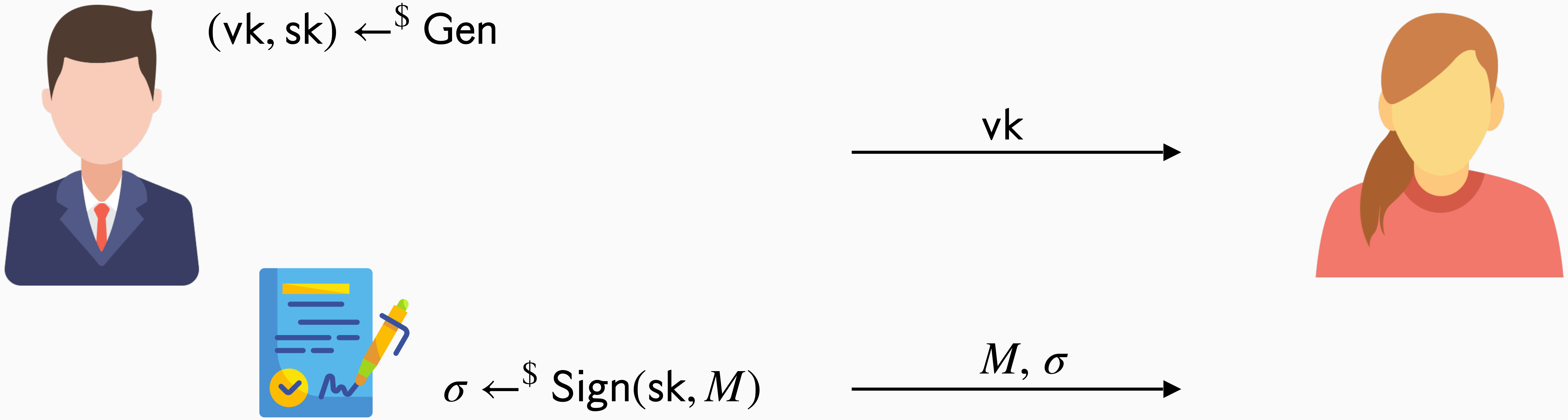
Syntax: A signature scheme Sig is described via algorithms (Gen, Sign, Vrfy).

$(\mathsf{vk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{Gen}$

$\xrightarrow{\mathsf{vk}}$

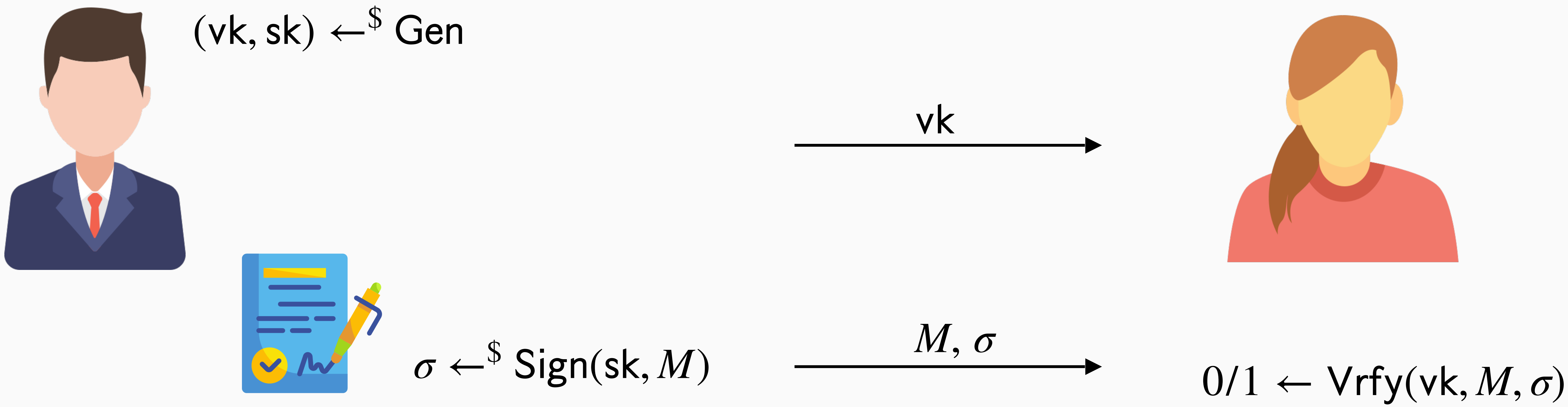
Digital Signatures

Syntax: A signature scheme Sig is described via algorithms (Gen, Sign, Vrfy).



Digital Signatures

Syntax: A signature scheme Sig is described via algorithms (Gen, Sign, Vrfy).



Unforgeability (Single-User)

Game $\mathbf{G}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$

Adversary $\mathcal{A}$

Unforgeability (Single-User)

Game $\mathbf{G}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$

$(\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}$

$\xrightarrow{\mathsf{vk}}$

Adversary $\mathcal{A}$

Unforgeability (Single-User)

Game $\mathbf{G}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{Gen}$

$\sigma \leftarrow^{\$} \mathsf{Sign}(\mathsf{sk}, M)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{M\}$

vk →

← $\textsc{Sign}$: M

σ →

Adversary $\mathcal{A}$

q queries

Unforgeability (Single-User)

Game $\mathbf{G}^{\text{uf-su}}_{\text{Sig}}$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{Gen}$

$\sigma \leftarrow^{\$} \mathsf{Sign}(\mathsf{sk}, M)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{M\}$

If $\mathsf{Vrfy}(\mathsf{vk}, M^\star, \sigma^\star) = 1$

and $M^\star \notin \mathcal{S}$:

  Return 1

Return 0

vk →

← SIGN: M

σ →

← $M^\star$, $\sigma^\star$

Adversary $\mathcal{A}$

q queries

Unforgeability (Single-User)

Game $\mathbf{G}^{\text{uf-su}}_{\text{Sig}}$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow^{\$} \mathsf{Gen}$

$\xrightarrow{\mathsf{vk}}$

$\xleftarrow{\textsc{Sign}: M}$

$\sigma \leftarrow^{\$} \mathsf{Sign}(\mathsf{sk}, M)$

$\xrightarrow{\sigma}$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{M\}$

Adversary $\mathcal{A}$

q queries

$\xleftarrow{M^\star, \sigma^\star}$

If $\mathsf{Vrfy}(\mathsf{vk}, M^\star, \sigma^\star) = 1$

and $M^\star \notin \mathcal{S}$:

Return 1

Return 0

$$\mathsf{Adv}^{\text{uf-su}}_{\text{Sig}}(\mathcal{A}) := \Pr[\mathbf{G}^{\text{uf-su}}_{\text{Sig}}(\mathcal{A}) = 1]$$

Unforgeability (Multi-User)

Game $\mathbf{G}^{\text{uf-mu-}n}_{\mathsf{Sig}}$

For $i \in \{1,\ldots,n\}$:

$(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow^{\$} \mathsf{Gen}$

$\sigma \leftarrow^{\$} \mathsf{Sign}(\mathsf{sk}_i, M)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, M)\}$

If $\mathsf{Vrfy}(\mathsf{vk}_{i^\star}, M^\star, \sigma^\star) = 1$

and $(i^\star, M^\star) \notin \mathcal{S}$:

Return 1

Return 0

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← Sign: i, M

σ →

← $i^\star$, $M^\star$, $\sigma^\star$

Adversary $\mathcal{A}$

q queries

$$\mathsf{Adv}^{\text{uf-mu-}n}_{\mathsf{Sig}}(\mathcal{A}) := \Pr[\mathbf{G}^{\text{uf-mu-}n}_{\mathsf{Sig}}(\mathcal{A}) = 1]$$

Unforgeability (Multi-User with Corruptions)

Game $\mathbf{G}^{\text{uf-muc-}n}_{\text{Sig}}$

For $i \in \{1,\ldots,n\}$:

$(\mathsf{vk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}$

$\sigma \xleftarrow{\$} \mathsf{Sign}(\mathsf{sk}_i, M)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, M)\}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$

If $\mathsf{Vrfy}(\mathsf{vk}_{i^\star}, M^\star, \sigma^\star) = 1$

and $(i^\star, M^\star) \notin \mathcal{S}$ and $i^\star \notin \mathcal{C}$:

Return 1

Return 0

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← Sign: i, M

σ →

← Corrupt: i

sk_i →

← $i^\star, M^\star, \sigma^\star$

Adversary $\mathcal{A}$

q queries

$< n$ queries

$$\mathsf{Adv}^{\text{uf-muc-}n}_{\text{Sig}}(\mathcal{A}) := \Pr[\mathbf{G}^{\text{uf-muc-}n}_{\text{Sig}}(\mathcal{A}) = 1]$$

Relations

	Multi-user	With corruptions
	$\mathsf{Adv}^{\mathsf{uf\text{-}mu\text{-}}n}_{\mathsf{Sig}}$	$\mathsf{Adv}^{\mathsf{uf\text{-}muc\text{-}}n}_{\mathsf{Sig}}$

Relations

	Multi-user $\mathsf{Adv}^{\text{uf-mu-}n}_{\mathsf{Sig}}$	With corruptions $\mathsf{Adv}^{\text{uf-muc-}n}_{\mathsf{Sig}}$
Type-I no better relations known than the general ones (e.g. RSA)	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$

Relations

	Multi-user $\mathsf{Adv}^{\text{uf-mu-}n}_{\mathsf{Sig}}$	**With corruptions** $\mathsf{Adv}^{\text{uf-muc-}n}_{\mathsf{Sig}}$
Type-I no better relations known than the general ones (e.g. RSA)	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$
Type-II mu-tight, but not under corruptions (e.g. Schnorr)	$\approx \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$

Relations

	Multi-user $\mathsf{Adv}^{\mathsf{uf\text{-}mu\text{-}}n}_{\mathsf{Sig}}$	**With corruptions** $\mathsf{Adv}^{\mathsf{uf\text{-}muc\text{-}}n}_{\mathsf{Sig}}$
Type-I no better relations known than the general ones (e.g. RSA)	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$
Type-II mu-tight, but not under corruptions (e.g. Schnorr)	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$
Type-III muc-tight (“special” constructions, e.g. [PKC:DGJL21])	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$

Relations

	Multi-user	With corruptions
	$\mathsf{Adv}^{\mathsf{uf\text{-}mu\text{-}}n}_{\mathsf{Sig}}$	$\mathsf{Adv}^{\mathsf{uf\text{-}muc\text{-}}n}_{\mathsf{Sig}}$
Type-I no better relations known than the general ones (e.g. RSA)	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$
Type-II mu-tight, but not under corruptions (e.g. Schnorr)	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$
Type-III muc-tight (“special” constructions, e.g. [PKC:DGJL21])	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$

mu-tight schemes seem to offer no advantage in the muc setting

Unforgeability (Multi-User with Corruptions)

Game $\mathbf{G}^{\text{uf-muc-}n}_{\text{Sig}}$

For $i \in \{1, \ldots, n\}$:

$(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow^{\$} \mathsf{Gen}$

$\sigma \leftarrow^{\$} \mathsf{Sign}(\mathsf{sk}_i, M)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, M)\}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$

If $\mathsf{Vrfy}(\mathsf{vk}_{i^\star}, M^\star, \sigma^\star) = 1$

and $(i^\star, M^\star) \notin \mathcal{S}$ and $i^\star \notin \mathcal{C}$:

Return 1

Return 0

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← Sign: i, M

σ →

← Corrupt: i

sk_i →

← $i^\star, M^\star, \sigma^\star$

Adversary $\mathcal{A}$

q queries

$< n$ queries

$\mathsf{Adv}^{\text{uf-muc-}n}_{\text{Sig}}(\mathcal{A}) := \Pr[\mathbf{G}^{\text{uf-muc-}n}_{\text{Sig}}(\mathcal{A}) = 1]$

Unforgeability (Multi-User with Corruptions)

Game $\mathbf{G}^{\text{uf-muc-}n}_{\text{Sig}}$

For $i \in \{1, \dots, n\}$:

$(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow^{\$} \mathsf{Gen}$

$\mathsf{vk}_1, \dots, \mathsf{vk}_n$ →

← Sign: i, M

$\sigma \leftarrow^{\$} \mathsf{Sign}(\mathsf{sk}_i, M)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, M)\}$

σ →

← Corrupt: i

$\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$

sk_i →

← $i^\star, M^\star, \sigma^\star$

If $\mathsf{Vrfy}(\mathsf{vk}_{i^\star}, M^\star, \sigma^\star) = 1$

and $(i^\star, M^\star) \notin \mathcal{S}$ and $i^\star \notin \mathcal{C}$:

Return 1

Return 0

Adversary $\mathcal{A}$

q queries

$< n$ queries

Always need to expect $n - 1$ corruptions

$$\mathsf{Adv}^{\text{uf-muc-}n}_{\mathsf{Sig}}(\mathcal{A}) := \Pr[\mathbf{G}^{\text{uf-muc-}n}_{\mathsf{Sig}}(\mathcal{A}) = 1]$$

Unforgeability (Multi-User with Corruptions)

Game $\mathbf{G}^{\text{uf-muc-}(n,c)}_{\text{Sig}}$

For $i \in \{1,\ldots,n\}$:

$(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow^{\$} \mathsf{Gen}$

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← $\textsc{Sign}$: i, M

$\sigma \leftarrow^{\$} \mathsf{Sign}(\mathsf{sk}_i, M)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, M)\}$

σ →

← $\textsc{Corrupt}$: i

$\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$

sk_i →

← $i^\star, M^\star, \sigma^\star$

If $\mathsf{Vrfy}(\mathsf{vk}_{i^\star}, M^\star, \sigma^\star) = 1$

and $(i^\star, M^\star) \notin \mathcal{S}$ and $i^\star \notin \mathcal{C}$:

Return 1

Return 0

Adversary $\mathcal{A}$

q queries

c queries ($c \ll n$)

$$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\text{Sig}}(\mathcal{A}) := \Pr[\mathbf{G}^{\text{uf-muc-}(n,c)}_{\text{Sig}}(\mathcal{A}) = 1]$$

Unforgeability (Multi-User with Corruptions)

Game $\mathbf{G}^{\text{uf-muc-}(n,c)}_{\text{Sig}}$

For $i \in \{1,\ldots,n\}$:

$(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow^{\$} \mathsf{Gen}$

$\sigma \leftarrow^{\$} \mathsf{Sign}(\mathsf{sk}_i, M)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, M)\}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$

If $\mathsf{Vrfy}(\mathsf{vk}_{i^\star}, M^\star, \sigma^\star) = 1$

and $(i^\star, M^\star) \notin \mathcal{S}$ and $i^\star \notin \mathcal{C}$:

Return 1

Return 0

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← Sign: i, M

σ →

← Corrupt: i

sk_i →

← $i^\star$, $M^\star$, $\sigma^\star$

Adversary $\mathcal{A}$

q queries

c queries ($c \ll n$)

More fine-grained view
(cp-muc, "corruption-parametrized")

$$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\text{Sig}}(\mathcal{A}) := \Pr[\mathbf{G}^{\text{uf-muc-}(n,c)}_{\text{Sig}}(\mathcal{A}) = 1]$$

cp-muc Theorem

Theorem (from su/mu to cp-muc):

Let n, c be integers s.t. $0 \leq c < n$. For any adversary $\mathcal{A}$ against uf-muc-(n, c) security of Sig, there exists an adversary $\mathcal{B}$ against uf-mu-m security of Sig s.t.

cp-muc Theorem

Theorem (from su/mu to cp-muc):

Let n, c be integers s.t. $0 \leq c < n$. For any adversary $\mathcal{A}$ against uf-muc-(n, c) security of Sig, there exists an adversary $\mathcal{B}$ against uf-mu-m security of Sig s.t.

$$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\mathsf{Sig}}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}^{\text{uf-mu-}m}_{\mathsf{Sig}}(\mathcal{B}) \qquad \text{where } e \approx 2.71,\ \ m = \left\lfloor \frac{n-1}{c-1} \right\rfloor$$

cp-muc Theorem

Theorem (from su/mu to cp-muc):

Let n, c be integers s.t. $0 \leq c < n$. For any adversary $\mathcal{A}$ against uf-muc-(n, c) security of Sig, there exists an adversary $\mathcal{B}$ against uf-mu-m security of Sig s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}muc\text{-}}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}mu\text{-}}m}(\mathcal{B}) \qquad \text{where } e \approx 2.71, \;\; m = \left\lfloor \frac{n-1}{c-1} \right\rfloor$$

For mu-tight secure schemes, there exists an adversary $\mathcal{B}'$ against uf-su security s.t.

cp-muc Theorem

Theorem (from su/mu to cp-muc):

Let n, c be integers s.t. $0 \leq c < n$. For any adversary $\mathcal{A}$ against uf-muc-(n, c) security of Sig, there exists an adversary $\mathcal{B}$ against uf-mu-m security of Sig s.t.

$$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\mathsf{Sig}}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}^{\text{uf-mu-}m}_{\mathsf{Sig}}(\mathcal{B}) \qquad \text{where } e \approx 2.71, \;\; m = \left\lfloor \frac{n-1}{c-1} \right\rfloor$$

For mu-tight secure schemes, there exists an adversary $\mathcal{B}'$ against uf-su security s.t.

$$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\mathsf{Sig}}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}(\mathcal{B}')$$

cp-muc Theorem

Theorem (from su/mu to cp-muc):

Let n, c be integers s.t. $0 \leq c < n$. For any adversary $\mathcal{A}$ against uf-muc-(n,c) security of Sig, there exists an adversary $\mathcal{B}$ against uf-mu-m security of Sig s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-muc-}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-mu-}m}(\mathcal{B}) \qquad \text{where } e \approx 2.71,\ \ m = \left\lfloor \frac{n-1}{c-1} \right\rfloor$$

For mu-tight secure schemes, there exists an adversary $\mathcal{B}'$ against uf-su security s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-muc-}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-su}}(\mathcal{B}')$$

main benefit for Type-II schemes

cp-muc Theorem

Theorem (from su/mu to cp-muc):

Let n, c be integers s.t. $0 \leq c < n$. For any adversary $\mathcal{A}$ against uf-muc-(n, c) security of Sig, there exists an adversary $\mathcal{B}$ against uf-mu-m security of Sig s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}muc\text{-}}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}mu\text{-}}m}(\mathcal{B}) \qquad \text{where } e \approx 2.71, \;\; m = \left\lfloor \frac{n-1}{c-1} \right\rfloor$$

assuming mu security for small number of users offers a non-trivial trade-off between su and muc

For mu-tight secure schemes, there exists an adversary $\mathcal{B}'$ against uf-su security s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}muc\text{-}}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}su}}(\mathcal{B}')$$

main benefit for Type-II schemes

cp-muc Theorem

Theorem (from su/mu to cp-muc):

Let n, c be integers s.t. $0 \leq c < n$. For any adversary $\mathcal{A}$ against uf-muc-(n, c) security of Sig, there exists an adversary $\mathcal{B}$ against uf-mu-m security of Sig s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-muc-}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-mu-}m}(\mathcal{B}) \qquad \text{where } e \approx 2.71, \; m = \left\lfloor \frac{n-1}{c-1} \right\rfloor$$

assuming mu security for small number of users offers a non-trivial trade-off between su and muc

Example:
$n = 100$ Million
$c = 100$ Thousand
$m = 999$

For mu-tight secure schemes, there exists an adversary $\mathcal{B}'$ against uf-su security s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-muc-}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-su}}(\mathcal{B}')$$

main benefit for Type-II schemes

cp-muc Theorem

Theorem (from su/mu to cp-muc):

Let n, c be integers s.t. $0 \leq c < n$. For any adversary $\mathcal{A}$ against uf-muc-(n, c) security of Sig, there exists an adversary $\mathcal{B}$ against uf-mu-m security of Sig s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-muc-}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-mu-}m}(\mathcal{B}) \qquad \text{where } e \approx 2.71, \ \ m = \left\lfloor \frac{n-1}{c-1} \right\rfloor$$

assuming mu security for small number of users offers a non-trivial trade-off between su and muc

Example:
$n = 100$ Million
$c = 100$ Thousand
$m = 999$

For mu-tight secure schemes, there exists an adversary $\mathcal{B}'$ against uf-su security s.t.

$$\mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-muc-}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-su}}(\mathcal{B}')$$

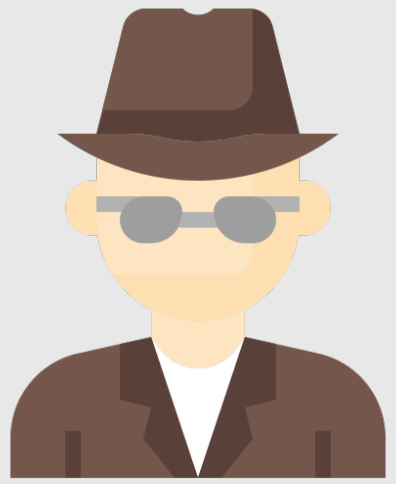
main benefit for Type-II schemes

Inspiration: Optimal bounds for FDH signatures [C:Coron01]

- Instead of losing a factor linear in the number of hash queries, reduction loses number of signing queries

cp-muc Theorem

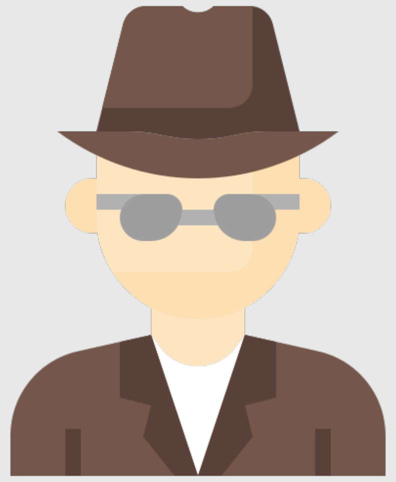
Refining and generalizing [C:Coron01]



cp-muc Theorem

Refining and generalizing [C:Coron01]

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_n$ →



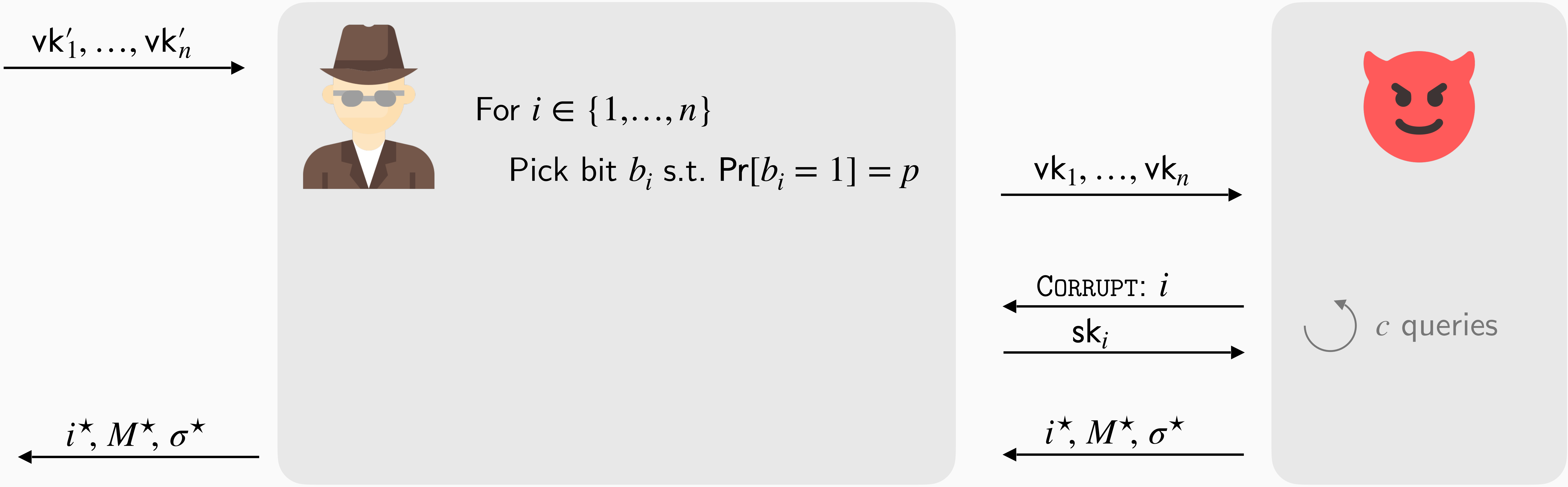
← $i^\star, M^\star, \sigma^\star$

cp-muc Theorem

Refining and generalizing [C:Coron01]

cp-muc Theorem

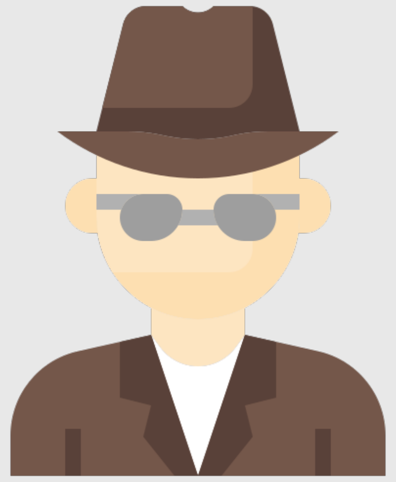
Refining and generalizing [C:Coron01]



cp-muc Theorem

Refining and generalizing [C:Coron01]

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_n$ →



For $i \in \{1, \ldots, n\}$

Pick bit b_i s.t. $\Pr[b_i = 1] = p$

If $b_i = 0$: $(\mathsf{vk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}$

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← CORRUPT: i

sk_i →

c queries

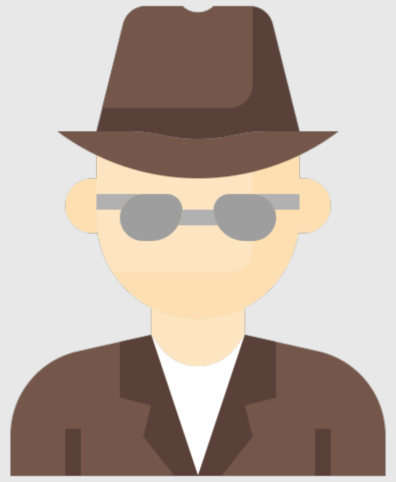
← $i^\star, M^\star, \sigma^\star$

← $i^\star, M^\star, \sigma^\star$

cp-muc Theorem

Refining and generalizing [C:Coron01]

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_n$ →



For $i \in \{1, \ldots, n\}$

Pick bit b_i s.t. $\Pr[b_i = 1] = p$

If $b_i = 0$: $(\mathsf{vk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}$

If $b_i = 1$: $\mathsf{vk}_i \leftarrow \mathsf{vk}'_i$

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← CORRUPT: i

sk_i →

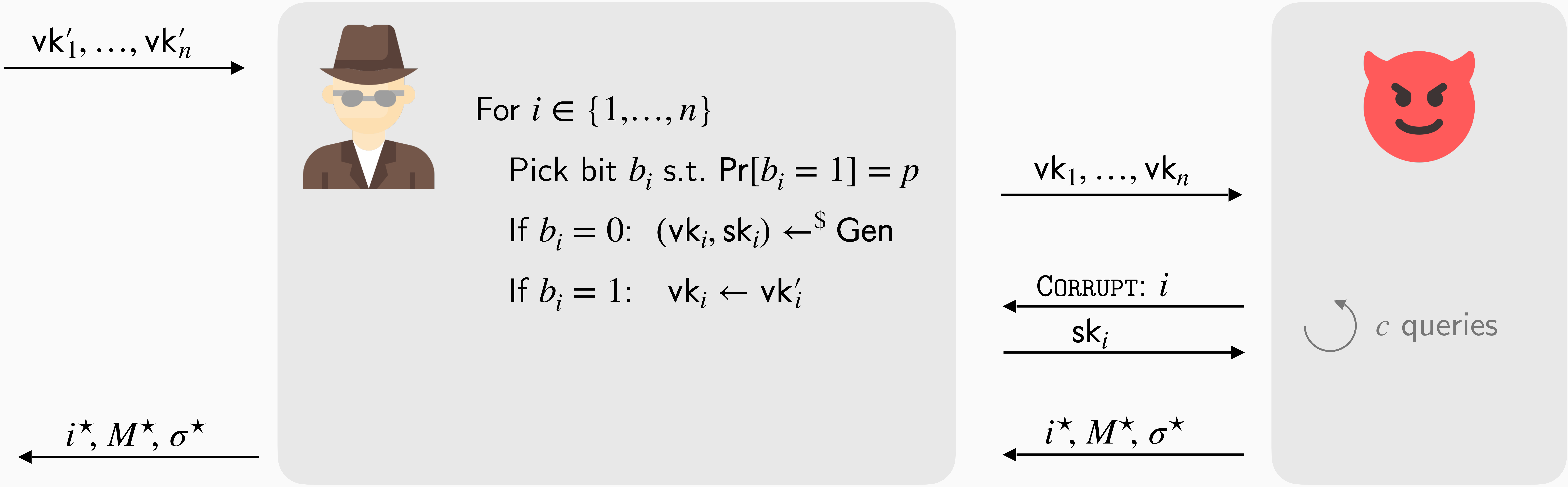
c queries

← $i^\star, M^\star, \sigma^\star$

← $i^\star, M^\star, \sigma^\star$

cp-muc Theorem

Refining and generalizing [C:Coron01]

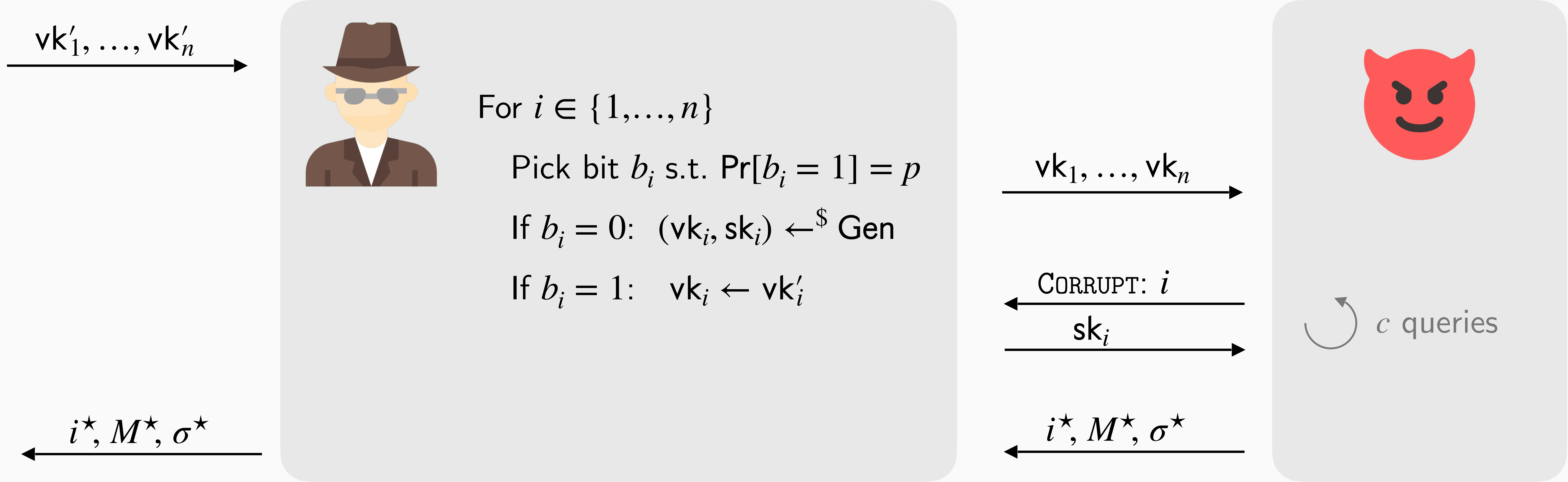


Reduction is successful if

- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

cp-muc Theorem

Refining and generalizing [C:Coron01]



Reduction is successful if

- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

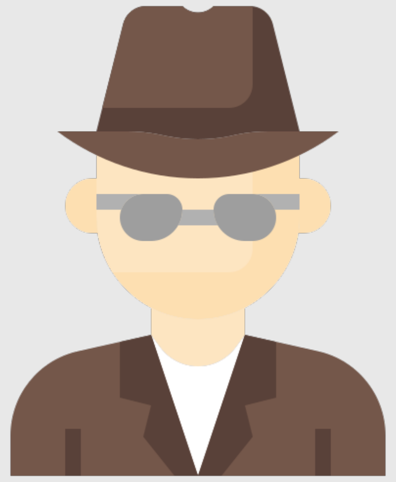
$$\mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-muc-}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\text{uf-mu-}n}(\mathcal{B})$$

cp-muc Theorem

Refining and generalizing [C:Coron01]

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_n$ →

We don't need n public keys!



For $i \in \{1,\ldots,n\}$

Pick bit b_i s.t. $\Pr[b_i = 1] = p$

If $b_i = 0$: $(\mathsf{vk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}$

If $b_i = 1$: $\mathsf{vk}_i \leftarrow \mathsf{vk}'_i$

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← CORRUPT: i

sk_i →

← $i^\star, M^\star, \sigma^\star$

c queries

← $i^\star, M^\star, \sigma^\star$

Reduction is successful if

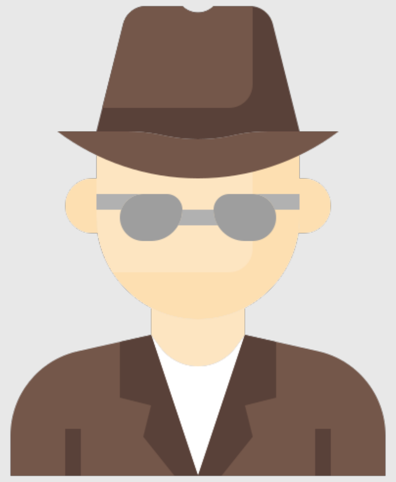
- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

$$\mathsf{Adv}^{\mathsf{uf\text{-}muc\text{-}}(n,c)}_{\mathsf{Sig}}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}^{\mathsf{uf\text{-}mu\text{-}}n}_{\mathsf{Sig}}(\mathcal{B})$$

cp-muc Theorem

Refining and generalizing [C:Coron01]

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_m$ →



For $i \in \{1, \ldots, n\}$

Pick bit b_i s.t. $\Pr[b_i = 1] = p$

If $b_i = 0$: $(\mathsf{vk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}$

If $b_i = 1$: Use next $\mathsf{vk}'_{i'}$

← $i^\star, M^\star, \sigma^\star$

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← CORRUPT: i

sk_i →

← $i^\star, M^\star, \sigma^\star$

c queries

Reduction is successful if

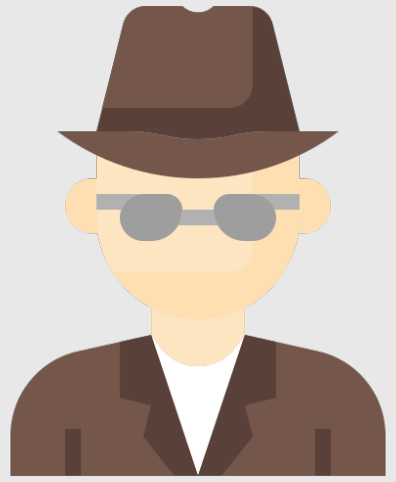
- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

$$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\mathsf{Sig}}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}^{\text{uf-mu-}n}_{\mathsf{Sig}}(\mathcal{B})$$

cp-muc Theorem

Refining and generalizing [C:Coron01]

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_m \rightarrow$



For $i \in \{1, \ldots, n\}$

Pick bit b_i s.t. $\Pr[b_i = 1] = p$

If $b_i = 0$: $(\mathsf{vk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}$

If $b_i = 1$: Use next $\mathsf{vk}'_{i'}$

This may fail for small m!

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n \rightarrow$

$\leftarrow$ CORRUPT: i

$\mathsf{sk}_i \rightarrow$

c queries

$\leftarrow i^\star, M^\star, \sigma^\star$

$\leftarrow i^\star, M^\star, \sigma^\star$

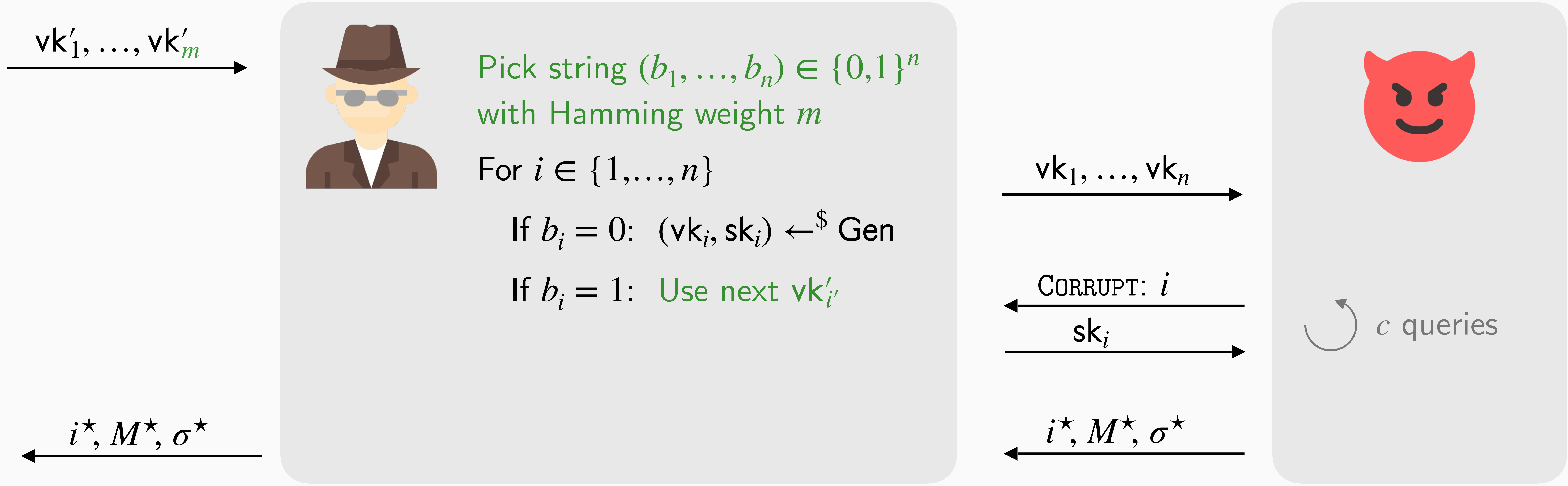
Reduction is successful if

- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

$$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\mathsf{Sig}}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}^{\text{uf-mu-}n}_{\mathsf{Sig}}(\mathcal{B})$$

cp-muc Theorem

Refining and generalizing [C:Coron01]



Reduction is successful if

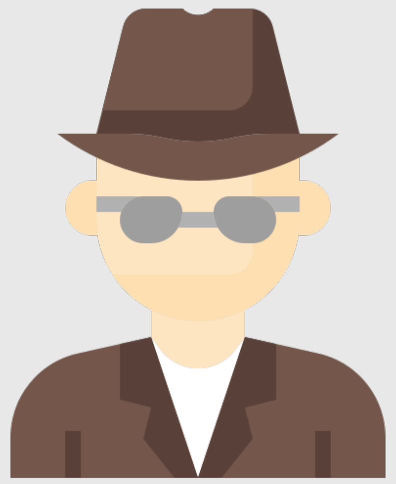
- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

$$\mathsf{Adv}^{\mathsf{uf\text{-}muc\text{-}}(n,c)}_{\mathsf{Sig}}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}^{\mathsf{uf\text{-}mu\text{-}}m}_{\mathsf{Sig}}(\mathcal{B})$$

cp-muc Theorem

Refining and generalizing [C:Coron01]

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_m$ →



Pick string $(b_1, \ldots, b_n) \in \{0,1\}^n$ with Hamming weight m

For $i \in \{1,\ldots,n\}$

If $b_i = 0$: $(\mathsf{vk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}$

If $b_i = 1$: Use next $\mathsf{vk}'_{i'}$

What is the optimal m?

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

CORRUPT: i ←

sk_i →

c queries

$i^\star, M^\star, \sigma^\star$ ←

← $i^\star, M^\star, \sigma^\star$

Reduction is successful if

- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

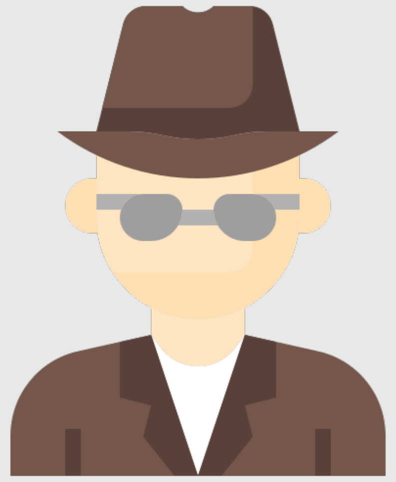
$$\mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}muc\text{-}}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}mu\text{-}}m}(\mathcal{B})$$

cp-muc Theorem

Refining and generalizing [C:Coron01]

This is captured by our abstraction of Hamming-weight determined samplers (via their success and error probability)

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_m$ →



Pick string $(b_1, \ldots, b_n) \in \{0,1\}^n$ with Hamming weight m

For $i \in \{1,\ldots,n\}$

If $b_i = 0$: $(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow^{\$} \mathsf{Gen}$

If $b_i = 1$: Use next $\mathsf{vk}'_{i'}$

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← CORRUPT: i

sk_i →

c queries

← $i^\star, M^\star, \sigma^\star$

← $i^\star, M^\star, \sigma^\star$

Reduction is successful if

- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

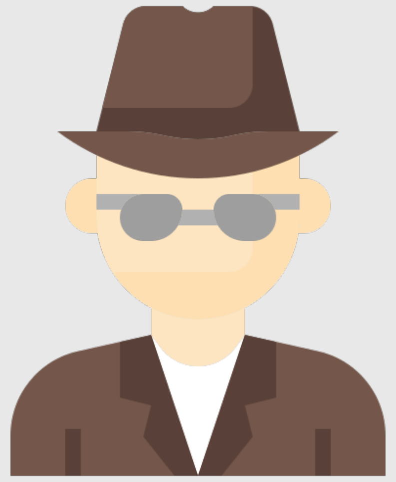
$$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\mathsf{Sig}}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}^{\text{uf-mu-}m}_{\mathsf{Sig}}(\mathcal{B})$$

cp-muc Theorem

Refining and generalizing [C:Coron01]

This is captured by our abstraction of Hamming-weight determined samplers (via their success and error probability)

$\mathsf{vk}'_1, \ldots, \mathsf{vk}'_m$ →



Pick string $(b_1, \ldots, b_n) \in \{0,1\}^n$ with Hamming weight m

For $i \in \{1,\ldots,n\}$

If $b_i = 0$: $(\mathsf{vk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}$

If $b_i = 1$: Use next $\mathsf{vk}'_{i'}$

← $i^\star, M^\star, \sigma^\star$

$\mathsf{vk}_1, \ldots, \mathsf{vk}_n$ →

← CORRUPT: i

sk_i →

← $i^\star, M^\star, \sigma^\star$

c queries

Reduction is successful if

- Corruption queries are only issued for users i s.t. $b_i = 0$
- Final solution is for a user $i^\star$ s.t. $b_{i^\star} = 1$

$$\mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}muc\text{-}}(n,c)}(\mathcal{A}) \leq e(c+1) \cdot \mathsf{Adv}_{\mathsf{Sig}}^{\mathsf{uf\text{-}mu\text{-}}m}(\mathcal{B})$$

for $m \approx n/c$

Relations

	Multi-user $\mathsf{Adv}^{\text{uf-mu-}n}_{\mathsf{Sig}}$	**With corruptions** $\mathsf{Adv}^{\text{uf-muc-}n}_{\mathsf{Sig}}$
Type-I no better relations known than the general ones (e.g. RSA)	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$
Type-II mu-tight, but not under corruptions (e.g. Schnorr)	$\approx \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$
Type-III muc-tight (“special” constructions, e.g. [PKC:DGJL21])	$\approx \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\approx \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$

Relations

	Multi-user	With corruptions	Parametrized
	$\mathsf{Adv}^{\text{uf-mu-}n}_{\mathsf{Sig}}$	$\mathsf{Adv}^{\text{uf-muc-}n}_{\mathsf{Sig}}$	$\mathsf{Adv}^{\text{uf-muc-}(n,c)}_{\mathsf{Sig}}$
Type-I no better relations known than the general ones (e.g. RSA)	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq e(c+1) \cdot \mathsf{Adv}^{\text{uf-mu-}m}_{\mathsf{Sig}}$
Type-II mu-tight, but not under corruptions (e.g. Schnorr)	$\approx \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\leq e(c+1) \cdot \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$
Type-III muc-tight (“special” constructions, e.g. [PKC:DGJL21])	$\approx \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\approx \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$	$\approx \mathsf{Adv}^{\text{uf-su}}_{\mathsf{Sig}}$

*$m = n/c$

Relations

	Multi-user $\mathsf{Adv}^{\mathsf{uf\text{-}mu\text{-}}n}_{\mathsf{Sig}}$	With corruptions $\mathsf{Adv}^{\mathsf{uf\text{-}muc\text{-}}n}_{\mathsf{Sig}}$	Parametrized $\mathsf{Adv}^{\mathsf{uf\text{-}muc\text{-}}(n,c)}_{\mathsf{Sig}}$
Type-I no better relations known than the general ones (e.g. RSA)	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\leq e(c+1) \cdot \mathsf{Adv}^{\mathsf{uf\text{-}mu\text{-}}m}_{\mathsf{Sig}}$
Type-II mu-tight, but not under corruptions (e.g. Schnorr)	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\leq n \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\leq e(c+1) \cdot \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$
Type-III muc-tight ("special" constructions, e.g. [PKC:DGJL21])	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$	$\approx \mathsf{Adv}^{\mathsf{uf\text{-}su}}_{\mathsf{Sig}}$

*$m = n/c$

Overview of our Results

Formal security specifications

- Syntax that translates into a single-user (su), multi-user (mu) and corruptions (muc) game

Hamming-weight determined samplers

- Technical tool that we introduce
- Essentially it determines how a (suitable) subset of users is picked

General cp-muc theorem (applies to all games which satisfy "locality" property)

- Basically all one-way (OW) games
- Indistinguishability (IND) games with independent challenge bits

Indirect applications of the cp-muc theorem (specialized results for "non-local" and "more advanced" games)

- IND-CCA with a single challenge bit across users (via FO, Hashed ElGamal)
- AKE protocols
- Selective opening security

We also give matching optimality (impossibility) results for a large class of games and schemes.

Conclusion

- In practice the number of corruptions is expected to be much smaller than the number of users.
- This was not reflected in models and thus concrete bounds for signing, encryption and key exchange.
- Our cp-muc framework gives a more fine-grained view and justifies standard parameter choices for many schemes.
  - It applies to Schnorr signatures, ElGamal-type encryption, and more.
- Tight muc security (Type-III schemes) is notoriously hard to achieve and we therefore suggest to focus on tight mu security (Type-II schemes).

Conclusion

- In practice the number of corruptions is expected to be much smaller than the number of users.
- This was not reflected in models and thus concrete bounds for signing, encryption and key exchange.
- Our cp-muc framework gives a more fine-grained view and justifies standard parameter choices for many schemes.
  - It applies to Schnorr signatures, ElGamal-type encryption, and more.
- Tight muc security (Type-III schemes) is notoriously hard to achieve and we therefore suggest to focus on tight mu security (Type-II schemes).

ePrint: `ia.cr/2024/1258`

Conclusion

- In practice the number of corruptions is expected to be much smaller than the number of users.
- This was not reflected in models and thus concrete bounds for signing, encryption and key exchange.
- Our cp-muc framework gives a more fine-grained view and justifies standard parameter choices for many schemes.
  - It applies to Schnorr signatures, ElGamal-type encryption, and more.
- Tight muc security (Type-III schemes) is notoriously hard to achieve and we therefore suggest to focus on tight mu security (Type-II schemes).

ePrint: `ia.cr/2024/1258`

Thank you!