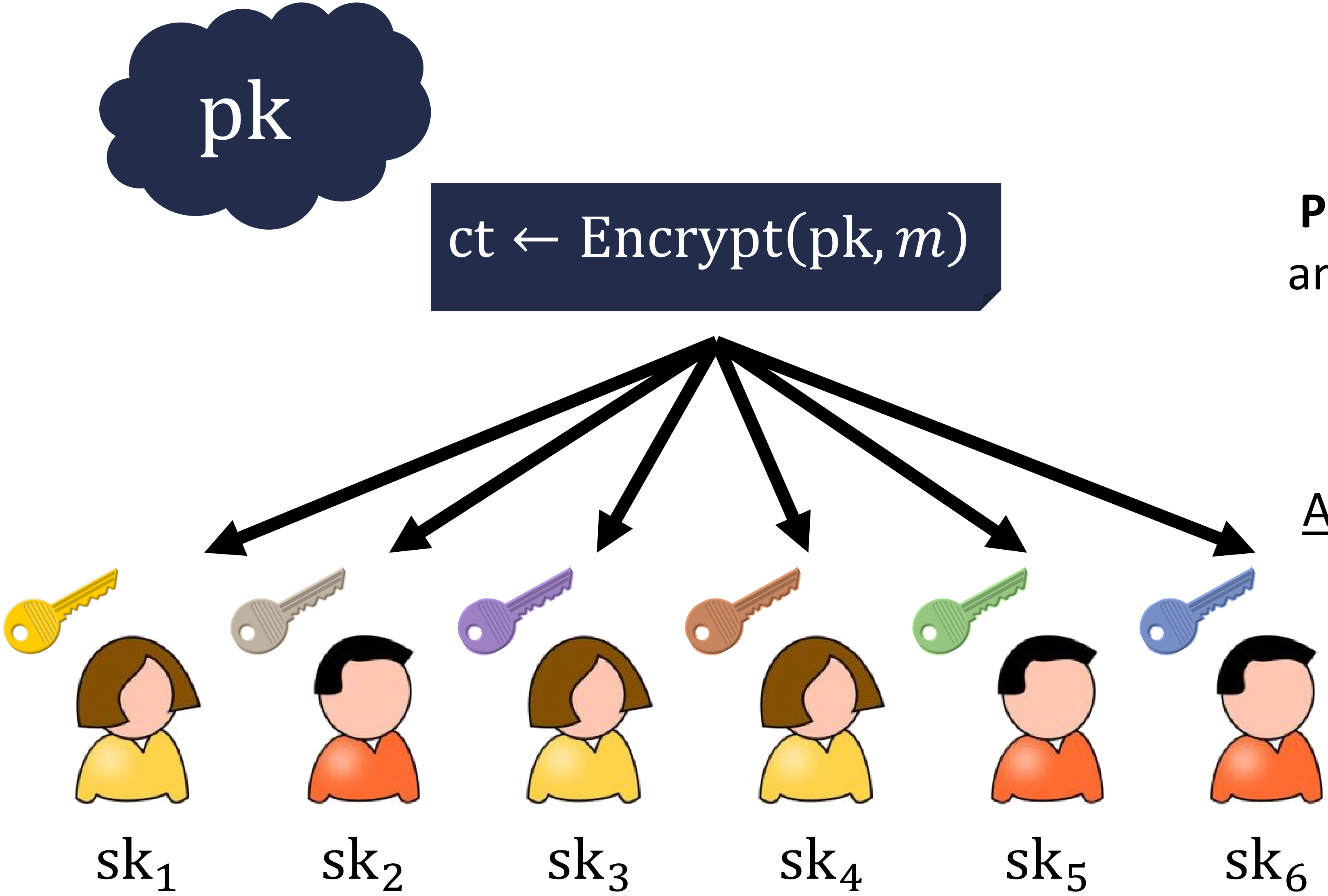
Collusion Resistant Trace-and-Revoke for Arbitrary Identities from Standard Assumptions

Sam Kim and David J. Wu

March 2021

Traitor Tracing

[CFN94]

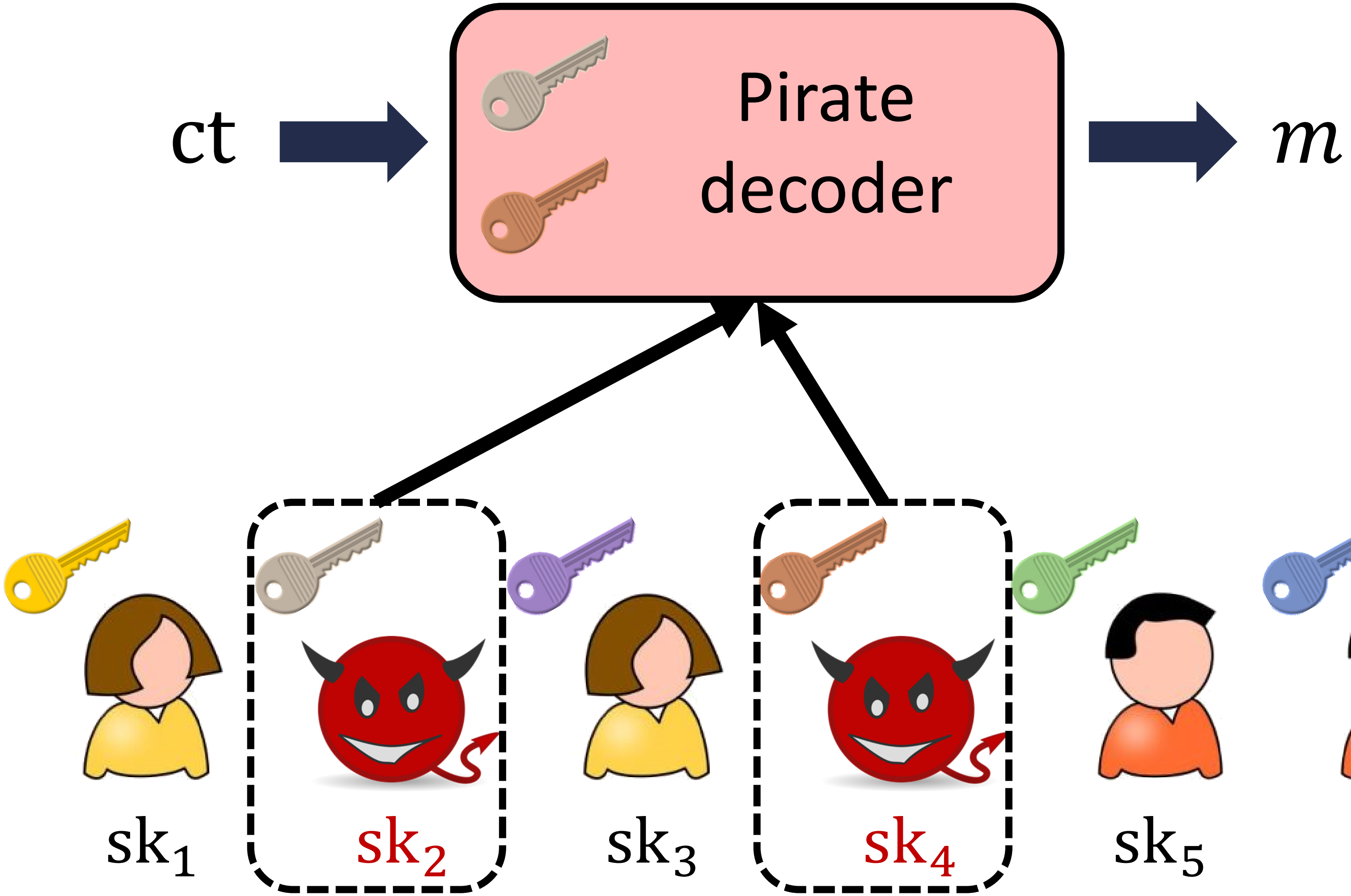


Public encryption:
anyone can encrypt

Any secret key sk_i
can decrypt ct

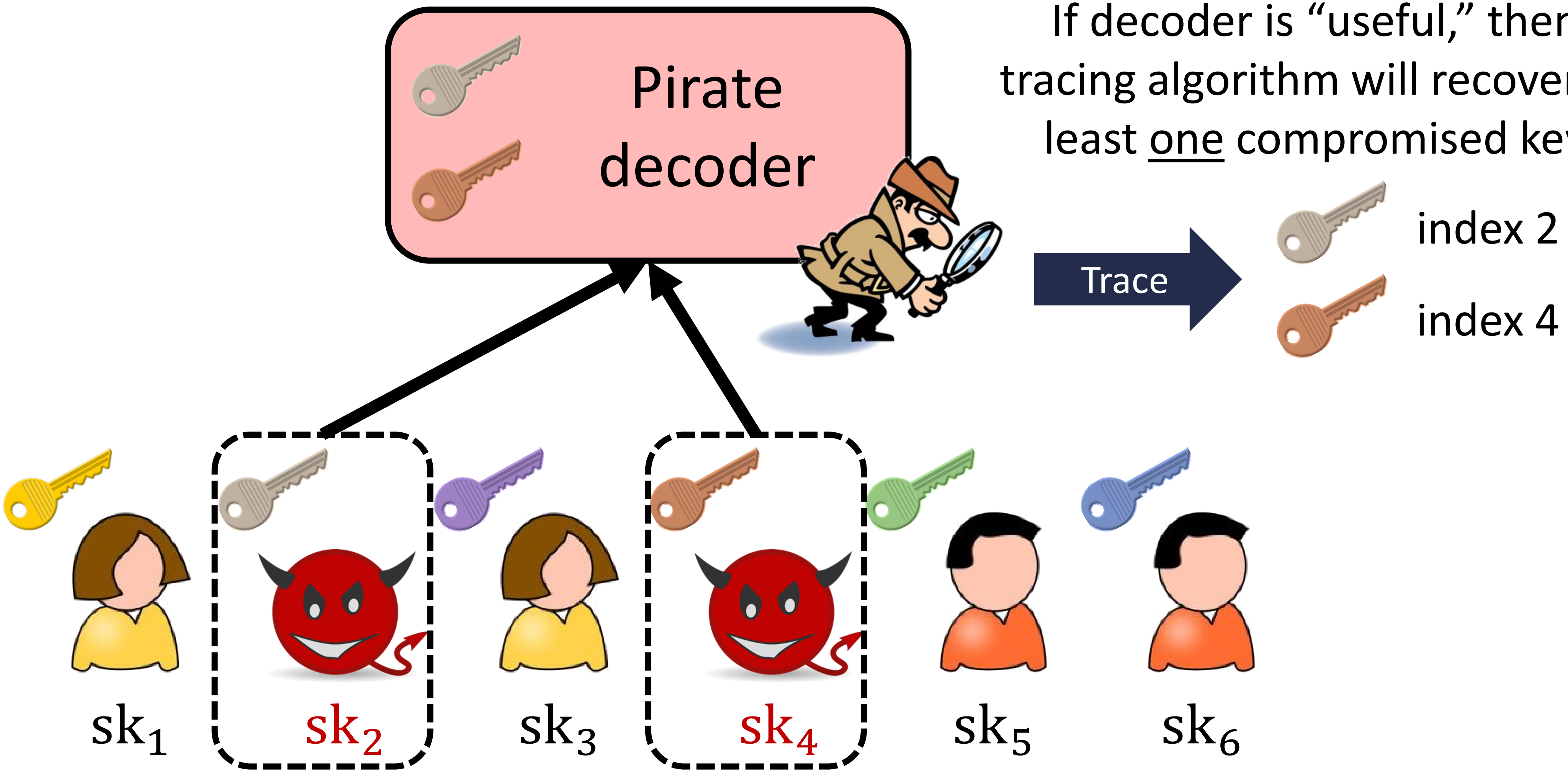
Traitor Tracing

[CFN94]



Traitor Tracing

[CFN94]



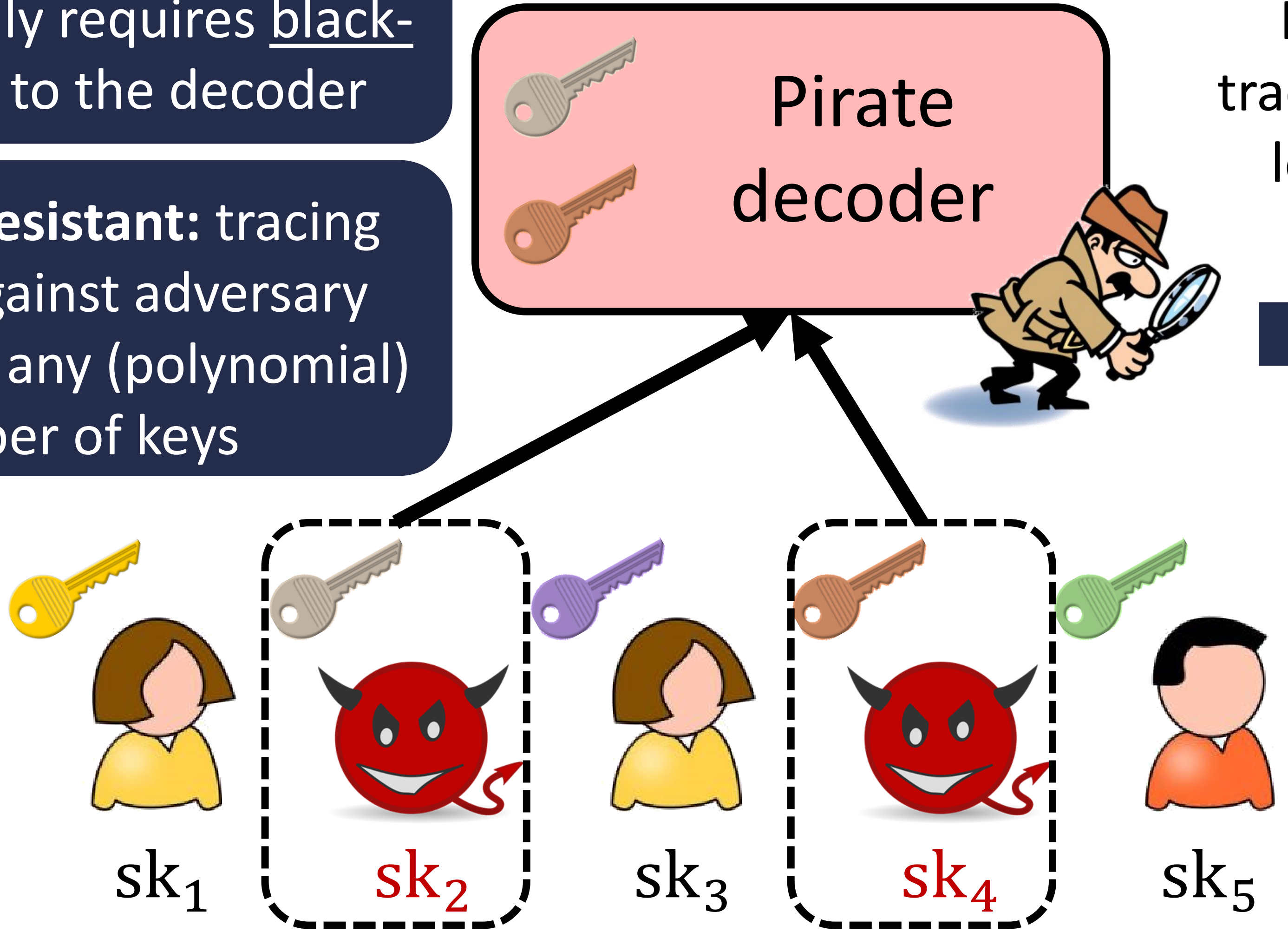
If decoder is "useful," then tracing algorithm will recover at least <u>one</u> compromised key

Traitor Tracing

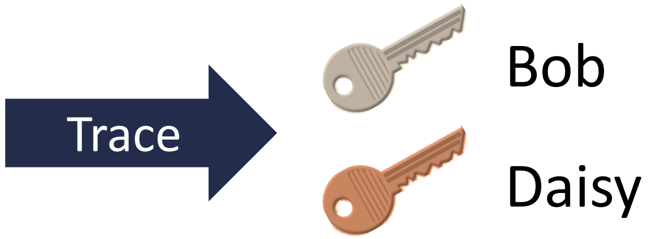
[CFN94]

Black-box tracing: tracing algorithm only requires black-box access to the decoder

Collusion-resistant: tracing possible against adversary who obtains any (polynomial) number of keys



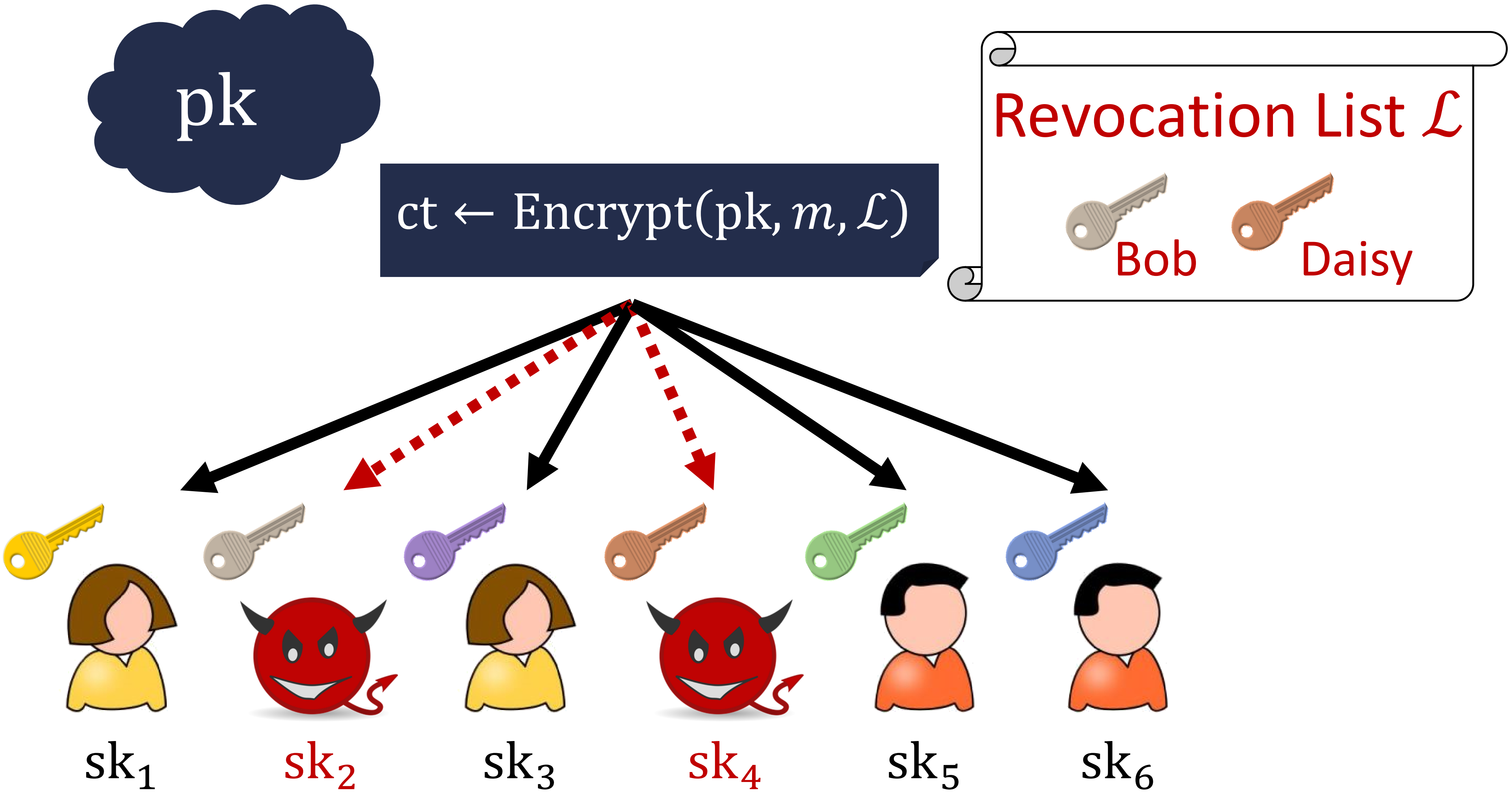
If decoder is "useful," then tracing algorithm will recover at least one compromised key



Identity-based traitor tracing: keys are associated with *arbitrary* identities (from an *exponential-size* space)

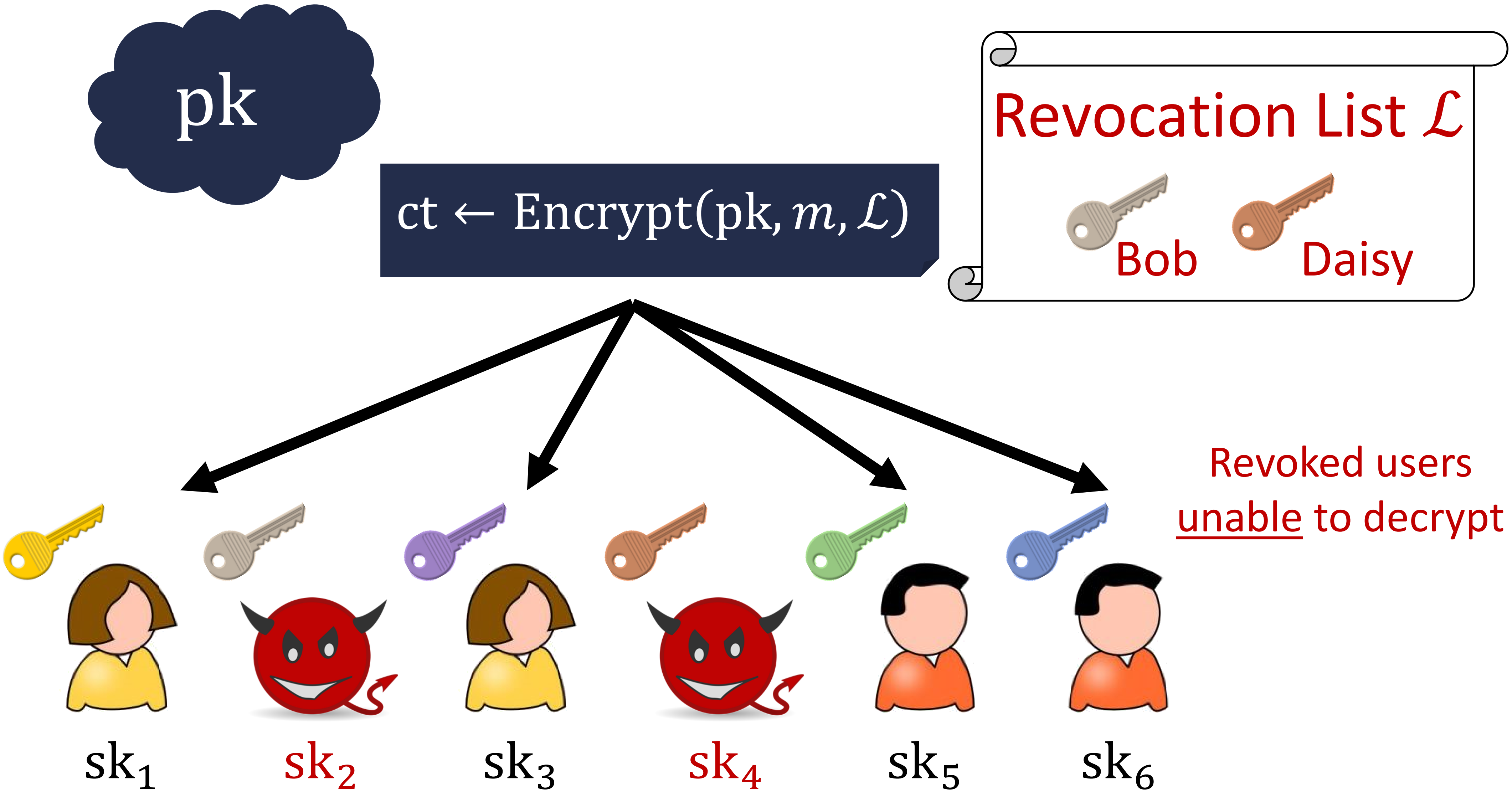
Trace and Revoke

[BW06]



Trace and Revoke

[BW06]



Identity-Based Trace and Revoke

[NWZ16]

Important: decoder only needs to distinguish between encryptions of two messages (i.e., break semantic security)

Formally:

- $\text{Setup}(1^\lambda) \to (\text{pp}, \text{msk})$
- $\text{KeyGen}(\text{msk}, \text{id})$ — generates secret key for $\text{id} \in \{0,1\}^{n(\lambda)}$
- $\text{Encrypt}(\text{pp}, m, \mathcal{L})$ — encrypts m with respect to revocation list $\mathcal{L}$
- $\text{Decrypt}(\text{sk}, \text{ct})$
- $\text{Trace}^{\mathcal{D}}(\text{msk}, m_0, m_1, \mathcal{L})$ — tracing algorithm has oracle access to a "good" decoder $\mathcal{D}$

$\mathcal{D}$ is good if $\Pr\left[b \leftarrow \{0,1\} : \mathcal{D}\big(\text{Encrypt}(\text{pp}, m_b, \mathcal{L})\big) = b\right] > \frac{1}{2} + \varepsilon$

This Work

Assuming sub-exponential hardness of LWE, there exists a fully collusion-resistant identity-based trace-and-revoke scheme

$|\mathrm{sk}| = n \cdot \mathrm{poly}(\lambda, \log n)$
$|\mathrm{ct}| = |m| + |\mathcal{L}| \cdot \mathrm{poly}(\lambda, \log n)$

m: message
n: bit-length of identity
$\mathcal{L}$: revocation list

Encryption algorithm is public-key
Tracing algorithm is secret-key

Existing construction of trace-and-revoke systems:

- Bounded collusion-resistant: [NWZ16, ABPSY17]
- Strong assumptions (e.g., iO or WE): [NWZ16, GVW19]
- Polynomial-size identity space: [BW06, GKSW10, GQWW19]

This Work

Assuming sub-exponential hardness of LWE, there exists a fully collusion-resistant identity-based trace-and-revoke scheme

General blueprint:

- Construct identity-based traitor tracing by combining ideas from Nishimaki et al. [NWZ16] and Goyal et al. [GKW18]
- Combine with combinatorial revocation approach of Naor et al. [NNL01] to obtain identity-based trace-and-revoke

Private Linear Broadcast Encryption

[BSW06]

Public encryption algorithm $\text{Encrypt}(\text{pk}, m)$

Secret keys are associated with index $i \in [N]$

Secret encryption algorithm to encrypt to an index t: $\text{Encrypt}(\text{sk}, t, m)$

x ciphertext

hidden index t

Can decrypt ciphertexts with indices $\text{id} \leq t$

Private Linear Broadcast Encryption

[BSW06]

Public encryption algorithm $\text{Encrypt}(\text{pk}, m)$

Secret keys are associated with index $i \in [N]$

Secret encryption algorithm to encrypt to an index t: $\text{Encrypt}(\text{sk}, t, m)$

Message hiding: ciphertexts with index 0 are semantically secure (given any collection of keys)

Index hiding: ciphertexts with index i and $i + 1$ are indistinguishable without key for $i + 1$

Private Linear Broadcast Encryption

[BSW06]

Public encryption algorithm $\text{Encrypt}(\text{pk}, m)$

Secret keys are associated with index $i \in [N]$

Secret encryption algorithm to encrypt to an index t: $\text{Encrypt}(\text{sk}, t, m)$

Message hiding: ciphertexts with index 0 are semantically secure (given any collection of keys)

Index hiding: ciphertexts with index i and $i + 1$ are indistinguishable without key for $i + 1$

"Strong attribute hiding:" indices are hidden even if the key successfully decrypts

Private Linear Broadcast Encryption

[BSW06]

Public encryption algorithm $\mathrm{Encrypt}(\mathrm{pk}, m)$

Secret keys are associated with index $i \in [N]$

Secret encryption algorithm to encrypt to an index t: $\mathrm{Encrypt}(\mathrm{sk}, t, m)$

Message hiding: ciphertexts with index 0 are semantically secure (given any collection of keys)

Index hiding: ciphertexts with index i and $i + 1$ are indistinguishable without key for $i + 1$

Indistinguishability: $\mathrm{Encrypt}(\mathrm{sk}, N, \cdot)$ is indistinguishable from $\mathrm{Encrypt}(\mathrm{pk}, \cdot)$

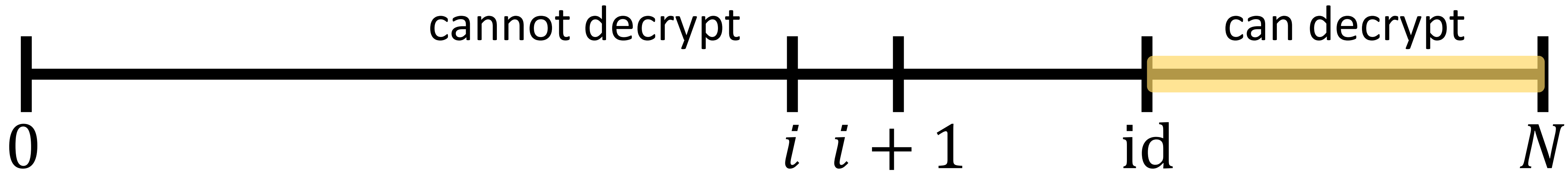
Private Linear Broadcast Encryption

[BSW06]

Tracing idea:

Assumption: Distinguisher D can break semantic security with advantage ε

Implication: There exists a jump in decoder advantage, and can only appear at id



Message Hiding

Ciphertexts with index 0 are semantically secure so decoder has negligible advantage

Index Hiding

Distinguishing advantage changes negligibly between i and $i+1$ when $\mathrm{id} \neq i+1$

Indistinguishability

Ciphertexts with index N are indistinguishable from real ciphertexts, so decoder has advantage ε

Private Linear Broadcast Encryption

[BSW06]

Public encryption algorithm $\mathrm{Encrypt}(\mathrm{pk}, m)$

Secret keys are associated with index $i \in [N]$

Secret encryption algorithm to encrypt to an index t: $\mathrm{Encrypt}(\mathrm{sk}, t, m)$

Message hiding: ciphertexts with index 0 are semantically secure (given any collection of keys)

Index hiding: ciphertexts with index i and j are indistinguishable without key for $i \leq \mathrm{id} < j$
(even for **exponentially-large** intervals)

Indistinguishability: $\mathrm{Encrypt}(\mathrm{sk}, N, \cdot)$ is

Enables tracing over exponential-size interval
(identity-based traitor tracing) [NWZ16]

PLBE from Mixed FE and ABE

[GKW18]

Mixed functional encryption (mixed FE):

Ciphertexts ct_f are associated with functions $f: \mathcal{X} \to \{0,1\}$

Decryption keys sk_x are associated with inputs $x \in \mathcal{X}$

Key-generation requires master secret key

$$\mathrm{Decrypt}(\mathrm{sk}_x, \mathrm{ct}_f) \to f(x) \in \{0,1\}$$

Two encryption algorithms:

- Public encryption: $\mathrm{PKEnc}(\mathrm{pp}) \to \mathrm{ct}$ (outputs encryption of all-ones function)
- Secret encryption: $\mathrm{SKEnc}(\mathrm{msk}, f) \to \mathrm{ct}_f$ (outputs encryption of function f)

PLBE from Mixed FE and ABE

[GKW18]

Mixed functional encryption (mixed FE):

Ciphertexts ct_f are associated with functions $f\colon \mathcal{X} \to \{0,1\}$

ct_{f_0} and ct_{f_1} are indistinguishable if $f_0(x) = f_1(x)$ for all keys x adversary has

Decryption keys sk_x are associated with inputs $x \in \mathcal{X}$

Key-generation requires master secret key

$$\mathrm{Decrypt}(\mathrm{sk}_x, \mathrm{ct}_f) \to f(x) \in \{0,1\}$$

Two encryption algorithms:

- Public encryption: $\mathrm{PKEnc}(\mathrm{pp}) \to \mathrm{ct}$
- Secret encryption: $\mathrm{SKEnc}(\mathrm{msk}, f) \to \mathrm{ct}_f$

Adversary who has secret key sk_x cannot distinguish $\mathrm{PKEnc}(\mathrm{pp})$ from $\mathrm{SKEnc}(\mathrm{msk}, f)$ whenever $f(x) = 1$

PLBE from Mixed FE and ABE

[GKW18]

Mixed functional encryption (mixed FE):

Ciphertexts ct_f are associated with functions $f\colon \mathcal{X} \to \{0,1\}$

ct_{f_0} and ct_{f_1} are indistinguishable if $f_0(x) = f_1(x)$ for all keys x adversary has

Decryption keys sk_x are associated with inputs $x \in \mathcal{X}$

Key-generation requires master secret key

$$\mathsf{Decrypt}(\mathsf{sk}_x, \mathsf{ct}_f) \to f(x) \in \{0,1\}$$

Two encryption algorithms:

- Public encryption: $\mathsf{PKEnc}(\mathsf{pp}) \to \mathsf{ct}$
- Secret encryption: $\mathsf{SKEnc}(\mathsf{msk}, f) \to \mathsf{ct}_f$

Selectively-secure mixed FE for circuits (with bounded ciphertext queries) known from LWE [GKW18, CVWWW19]

PLBE from Mixed FE and ABE

[GKW18]

Attribute-based encryption (ABE):

Ciphertexts $\mathsf{ct}_{x,m}$ are associated with public attribute $x \in \mathcal{X}$ and a message m

Encryption is public operation

Decryption keys sk_f are associated with predicate $f: \mathcal{X} \to \{0,1\}$

Key-generation requires master secret key

$$\mathrm{Decrypt}(\mathsf{sk}_f, \mathsf{ct}_{x,m}) \to \begin{cases} m, & f(x) = 1 \\ \perp, & f(x) = 0 \end{cases}$$

Selectively-secure ABE for circuits known from LWE [GVW13, BGGHNSVV14]

PLBE from Mixed FE and ABE

[GKW18]

$\text{KeyGen}(\text{sk}, \text{id})$

MFE secret key for id

Can decrypt ciphertexts with indices $\text{id} \leq t$

ABE key for function f

$f(\cdot) = \text{MFE.Decrypt}(\text{MFE.sk}_{\text{id}}, \cdot)$

$\text{Encrypt}(\text{sk}, t, m)$

MFE ciphertext for comparison function g_t

$$g_t(\text{id}) = \begin{cases} 1, & \text{id} \leq t \\ 0, & \text{id} > t \end{cases}$$

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute MFE.ct_{g_t}

Correctness: If $\text{id} \leq t$, then $g_t(\text{id}) = 1$, so ABE decryption succeeds

PLBE from Mixed FE and ABE

[GKW18]

$\text{KeyGen}(\text{sk}, \text{id})$

MFE secret key for id

Can decrypt ciphertexts with indices $\text{id} \leq t$

ABE key for function f

$f(\cdot) = \text{MFE.Decrypt}(\text{MFE.sk}_{\text{id}}, \cdot)$

$\text{Encrypt}(\text{sk}, t, m)$

MFE ciphertext for comparison function g_t

$$g_t(\text{id}) = \begin{cases} 1, & \text{id} \leq t \\ 0, & \text{id} > t \end{cases}$$

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute MFE.ct_{g_t}

Message hiding: Ciphertexts with index 0 are semantically secure (given any collection of keys)

If $t = 0$, then $g_t(\text{id}) = 0$ for all id, so semantic security by ABE security

PLBE from Mixed FE and ABE

[GKW18]

$\mathrm{KeyGen}(\mathrm{sk}, \mathrm{id})$

MFE secret key for id

Can decrypt ciphertexts with indices $\mathrm{id} \le t$

ABE key for function f

$f(\cdot) = \mathrm{MFE.Decrypt}(\mathrm{MFE.sk}_{\mathrm{id}}, \cdot)$

$\mathrm{Encrypt}(\mathrm{sk}, t, m)$

MFE ciphertext for comparison function g_t

$$g_t(\mathrm{id}) = \begin{cases} 1, & \mathrm{id} \le t \\ 0, & \mathrm{id} > t \end{cases}$$

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute $\mathrm{MFE.ct}_{g_t}$

Index hiding: Ciphertexts with index i and $i+1$ are indistinguishable without key for $i+1$

$\mathrm{MFE.ct}_{g_i}$ and $\mathrm{MFE.ct}_{g_{i+1}}$ indistinguishable without $\mathrm{MFE.sk}_{i+1}$

PLBE from Mixed FE and ABE

[GKW18]

$\text{KeyGen}(\text{sk}, \text{id})$

MFE secret key for id

Can decrypt ciphertexts with indices $\text{id} \leq t$

ABE key for function f

$f(\cdot) = \text{MFE.Decrypt}(\text{MFE.sk}_{\text{id}}, \cdot)$

$\text{Encrypt}(\text{sk}, t, m)$

MFE ciphertext for comparison function g_t

$$g_t(\text{id}) = \begin{cases} 1, & \text{id} \leq t \\ 0, & \text{id} > t \end{cases}$$

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute MFE.ct_{g_t}

Indistinguishability: $\text{Encrypt}(\text{sk}, N, \cdot)$ is indistinguishable from $\text{Encrypt}(\text{pk}, \cdot)$

$g_N(\text{id}) = 1$ for all id; follows by MFE public/secret indistinguishability

PLBE from Mixed FE and ABE

[GKW18]

KeyGen(sk, id)

MFE secret key for id

Can decrypt ciphertexts with indices $\mathrm{id} \leq t$

ABE key for function f

$f(\cdot) = \mathrm{MFE.Decrypt}(\mathrm{MFE.sk_{id}}, \cdot)$

Encrypt(sk, t, m)

MFE ciphertext for comparison function g_t

$$g_t(\mathrm{id}) = \begin{cases} 1, & \mathrm{id} \leq t \\ 0, & \mathrm{id} > t \end{cases}$$

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute $\mathrm{MFE.ct}_{g_t}$

[GKW18]: Instantiate mixed FE + selectively-secure ABE from polynomial hardness of LWE

⇒ PLBE for polynomial number of identities

⇒ Traitor tracing for polynomial number of identities from polynomial hardness of LWE

PLBE from Mixed FE and ABE

[GKW18]

$\mathsf{KeyGen}(\mathsf{sk}, \mathsf{id})$

MFE secret key for id

Can decrypt ciphertexts with indices $\mathsf{id} \leq t$

ABE key for function f

$f(\cdot) = \mathsf{MFE.Decrypt}(\mathsf{MFE.sk}_{\mathsf{id}}, \cdot)$

$\mathsf{Encrypt}(\mathsf{sk}, t, m)$

MFE ciphertext for comparison function g_t

$$g_t(\mathsf{id}) = \begin{cases} 1, & \mathsf{id} \leq t \\ 0, & \mathsf{id} > t \end{cases}$$

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute $\mathsf{MFE.ct}_{g_t}$

Complexity leveraging: Instantiate mixed FE + adaptively-secure ABE from sub-exponential hardness of LWE

⇒ PLBE for super-polynomial number of identities

PLBE from Mixed FE and ABE

[GKW18]

$\mathsf{KeyGen}(\mathsf{sk}, \mathsf{id})$

MFE secret key for id

Can decrypt ciphertexts with indices $\mathsf{id} \leq t$

ABE key for function f

$f(\cdot) = \mathsf{MFE.Decrypt}(\mathsf{MFE.sk}_{\mathsf{id}}, \cdot)$

$\mathsf{Encrypt}(\mathsf{sk}, t, m)$

MFE ciphertext for comparison function g_t

$$g_t(\mathsf{id}) = \begin{cases} 1, & \mathsf{id} \leq t \\ 0, & \mathsf{id} > t \end{cases}$$

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute $\mathsf{MFE.ct}_{g_t}$

Using tracing algorithm of [NWZ16]

Complexity leveraging: Instantiate mixed FE + adapt ... ential hardness of LWE

⇒ PLBE for super-polynomia ...

⇒ Traitor tracing for super-polynomial number of identities from sub-exponential LWE

Secret-Key Predicate Encryption

$\text{KeyGen}(\text{sk}, \text{id})$

MFE secret key for id

$\text{Encrypt}(\text{sk}, t, m)$

MFE ciphertext for comparison function g_t

$$g_t(\text{id}) = \begin{cases} 1, & \text{id} \leq t \\ 0, & \text{id} > t \end{cases}$$

Can decrypt ciphertexts with indices id $\leq t$

ABE key for function f

$f(\cdot) = \text{MFE.Decrypt}(\text{MFE.sk}_{\text{id}}, \cdot)$

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute MFE.ct_{g_t}

Can view this more generally as a
secret-key ciphertext-policy predicate encryption scheme with public broadcast

Secret-Key Predicate Encryption

$\text{KeyGen}(\text{msk}, x)$

MFE secret key for x

x is an attribute

Can decrypt ciphertexts ct_g where $g(x) = 1$

ABE key for function f

$f(\text{ct}_g) = \text{MFE.Decrypt}(\text{MFE.sk}_x, \text{ct}_g)$

$\text{Encrypt}(\text{msk}, g, m)$

MFE ciphertext for function g

g encodes the decryption policy

Public encryption: encrypt using public MFE encryption

ABE encryption of m with attribute MFE.ct_g

Can view this more generally as a
secret-key ciphertext-policy predicate encryption scheme with public broadcast

Revocable Predicate Encryption

Goal: allow encryption to take in a revocation list $\mathcal{L}$ of identities (decryption keys associated with identities)

MFE ciphertext for function g

ABE encryption of m with attribute $\mathrm{MFE.ct}_g$

Attempt: embed $\mathcal{L}$ as part of the ciphertext decryption policy and id with the key

$$g_{\mathcal{L}}(x, \mathrm{id}) = 1 \text{ if and only if } g(x) = 1 \wedge \mathrm{id} \notin \mathcal{L}$$

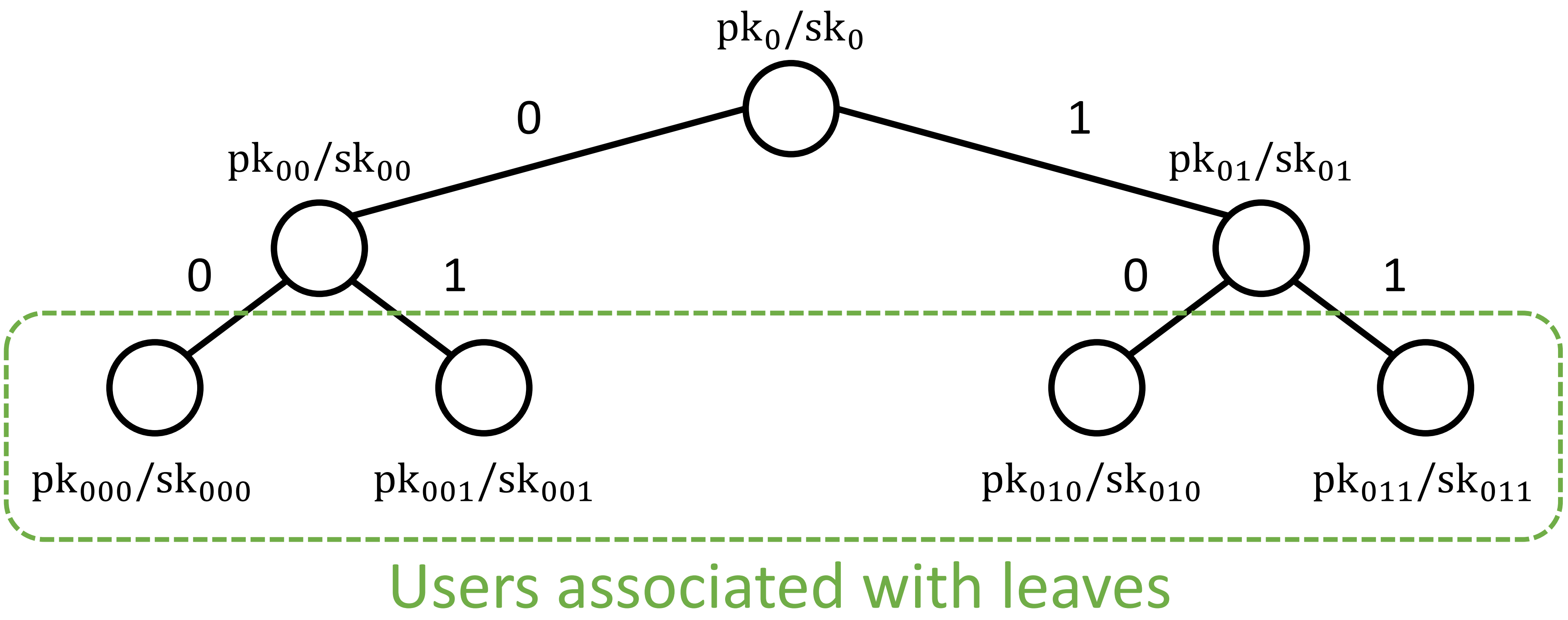
Problem: Public encryption algorithm only supports *broadcast* (strong attribute-hiding *public-key* predicate encryption equivalent to functional encryption)

Problem: Length of revocation list is *a priori* unbounded (incompatible with MFE for circuits)

Combinatoric Approach to Revocation

[NNL01]

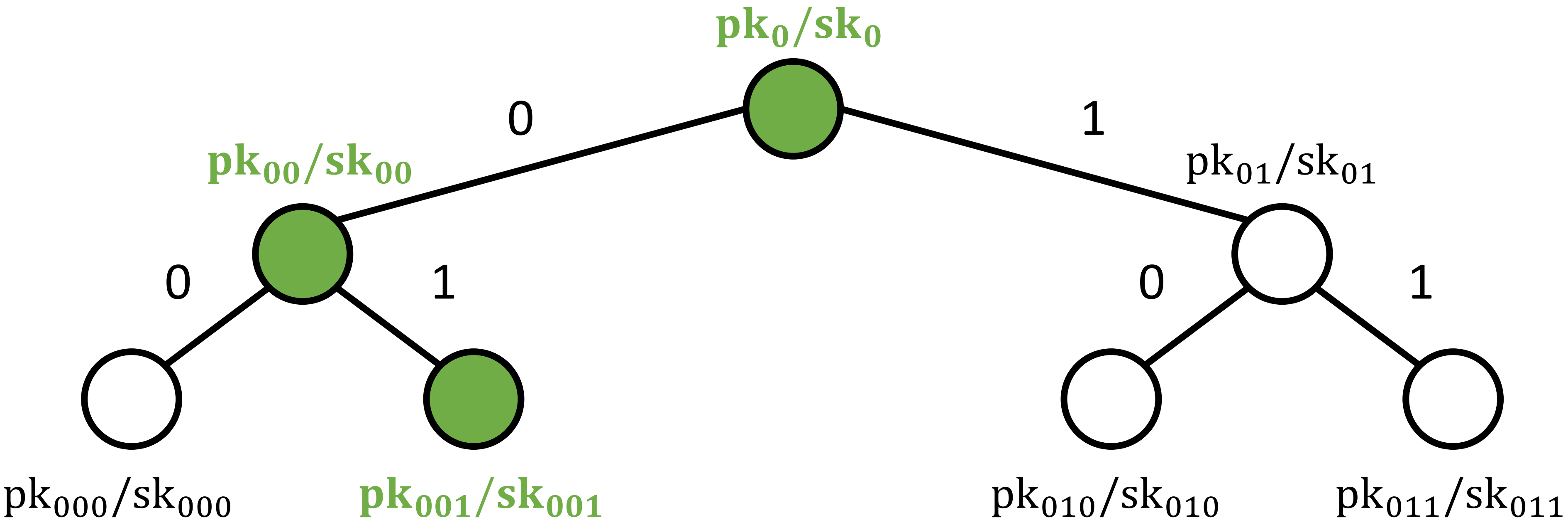
[NNL01]: Combinatoric approach for revocation based on subset-cover set systems



Combinatoric Approach to Revocation

[NNL01]

[NNL01]: Combinatoric approach for revocation based on subset-cover set systems



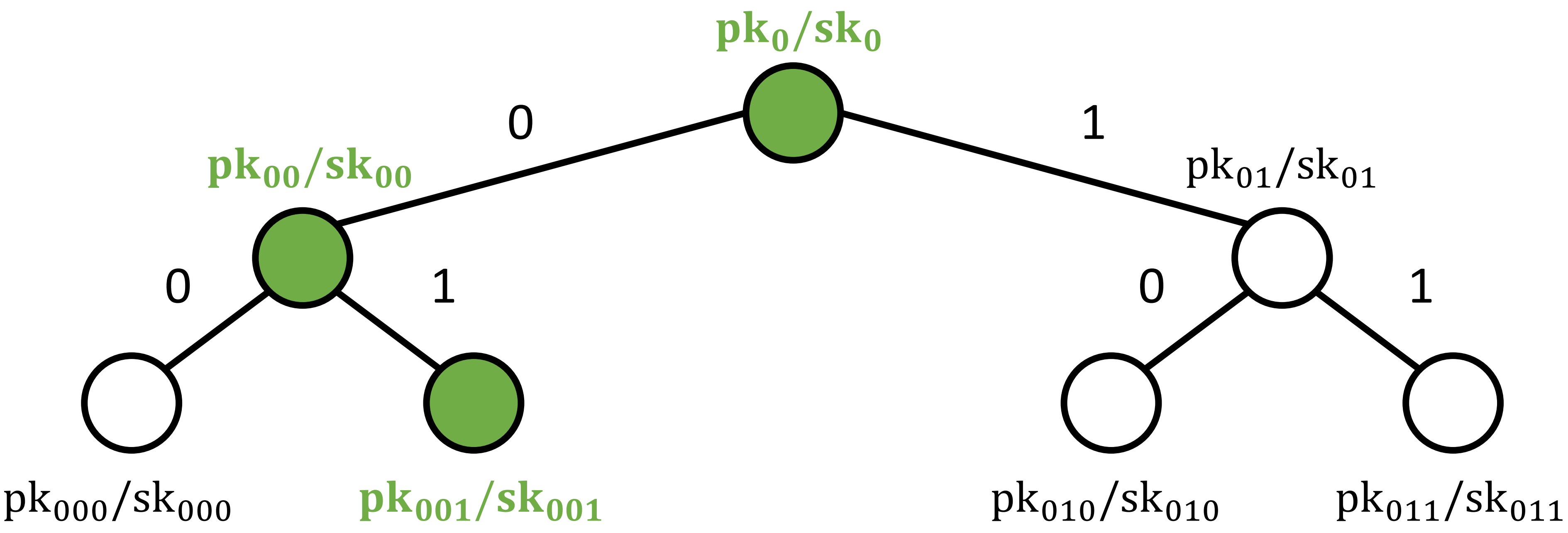
Secret key for user: all secret keys along the path

$$\mathrm{sk} = \{\mathrm{sk}_0, \mathrm{sk}_{00}, \mathrm{sk}_{001}\}$$

Combinatoric Approach to Revocation

[NNL01]

[NNL01]: Combinatoric approach for revocation based on subset-cover set systems

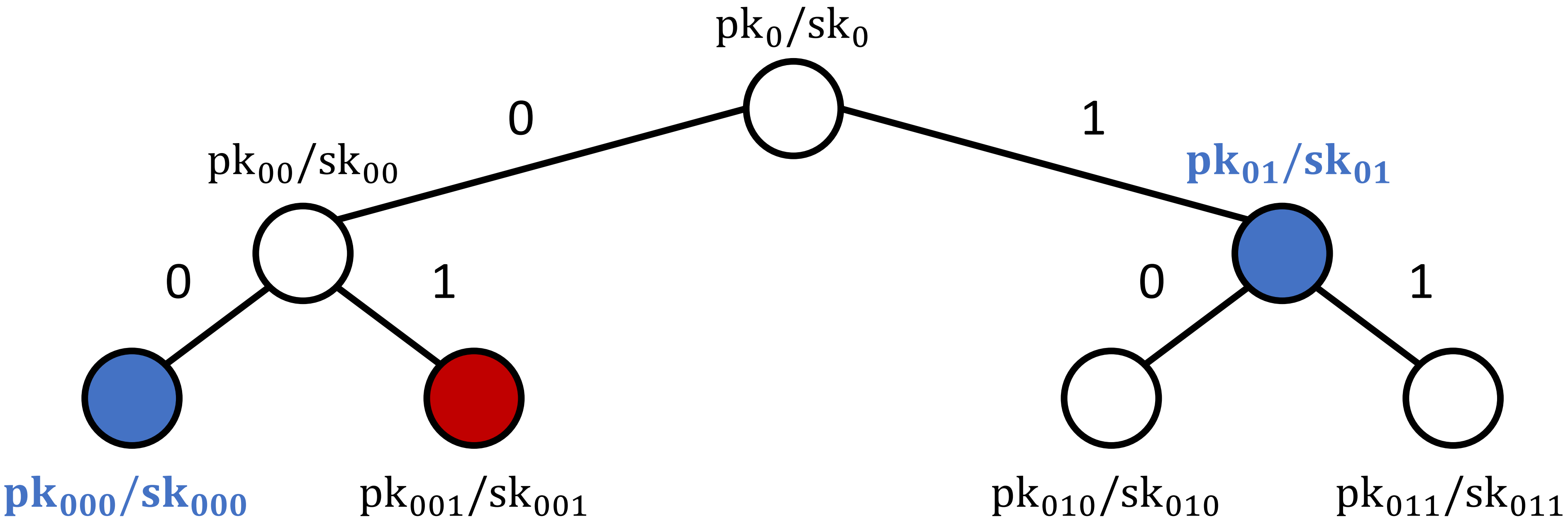


Encrypting to all users: encrypt under root key pk_0

Combinatoric Approach to Revocation

[NNL01]

[NNL01]: Combinatoric approach for revocation based on subset-cover set systems

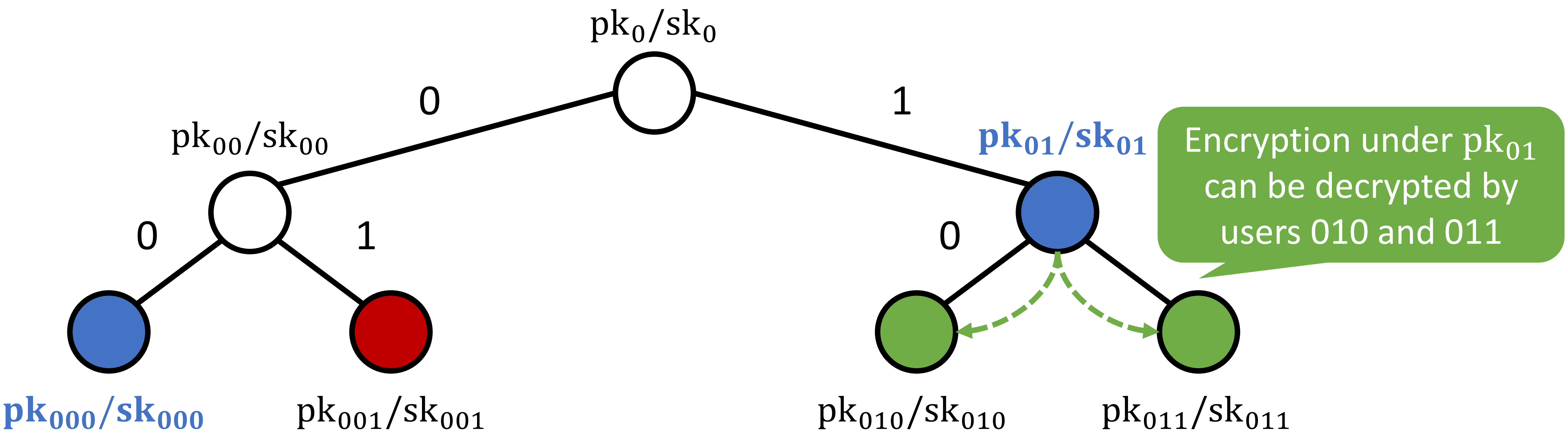


Revocation: encrypt under subset that excludes revoked users

Combinatoric Approach to Revocation

[NNL01]

[NNL01]: Combinatoric approach for revocation based on subset-cover set systems

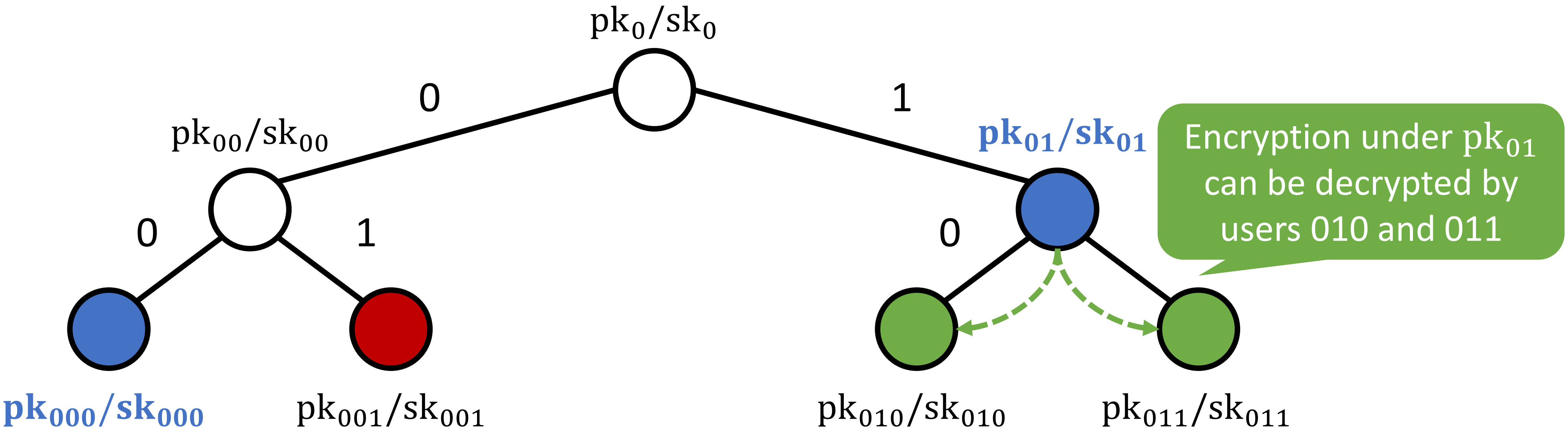


Revocation: encrypt under subset that excludes revoked users

Combinatoric Approach to Revocation

[NNL01]

[NNL01]: Combinatoric approach for revocation based on subset-cover set systems

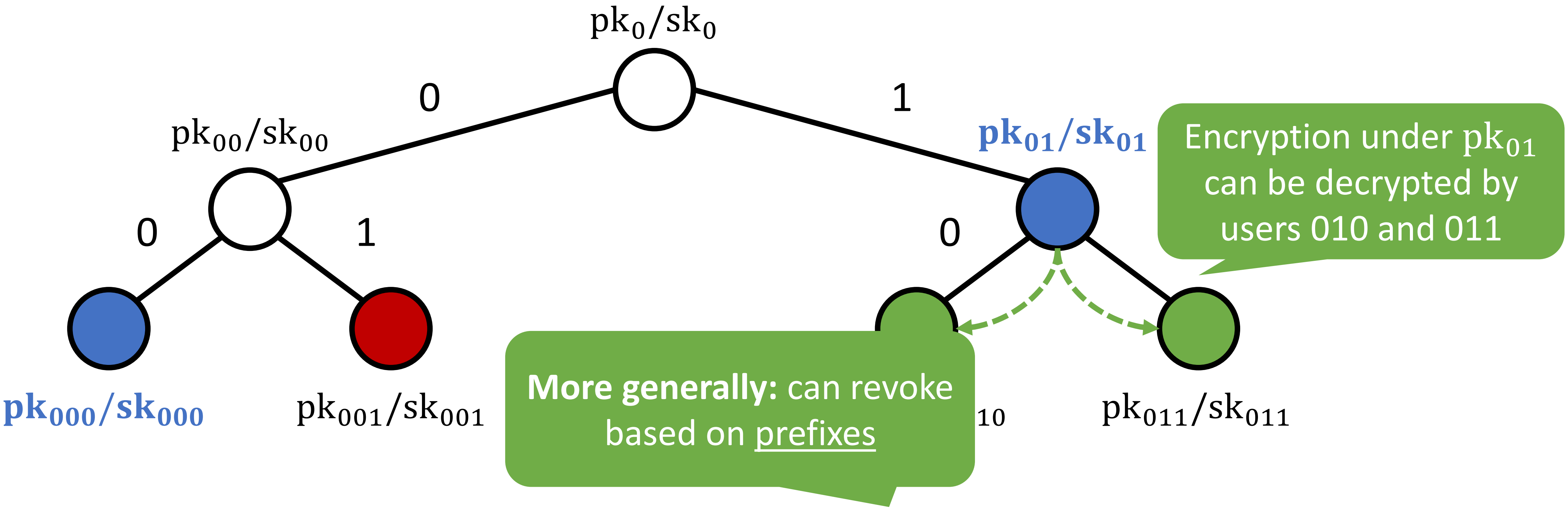


Generally: ciphertext consists of $O(\log|\mathcal{L}|)$ encryptions

Combinatoric Approach to Revocation

[NNL01]

[NNL01]: Combinatoric approach for revocation based on subset-cover set systems

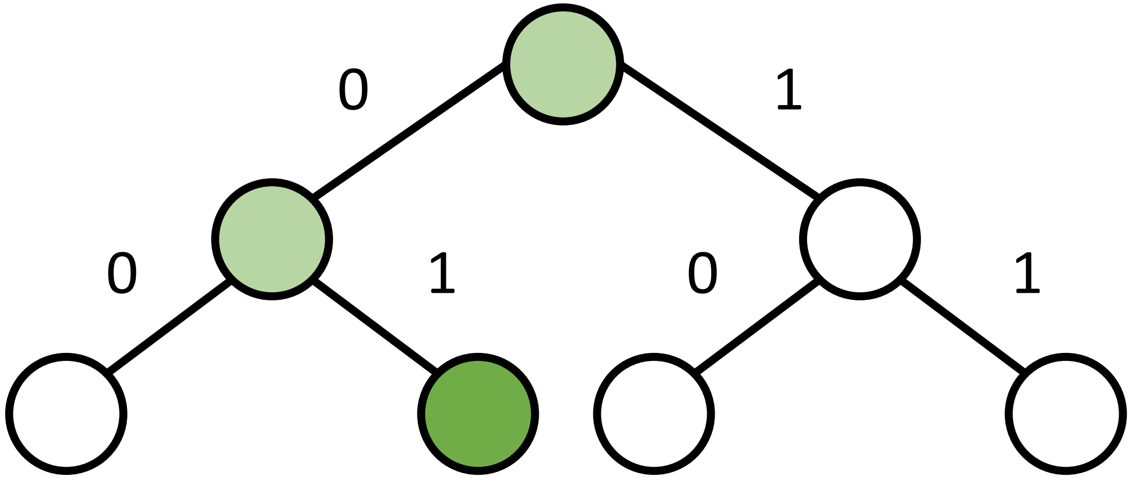


Generally: ciphertext consists of $O(|\mathcal{L}| \log|\mathcal{L}|)$ encryptions

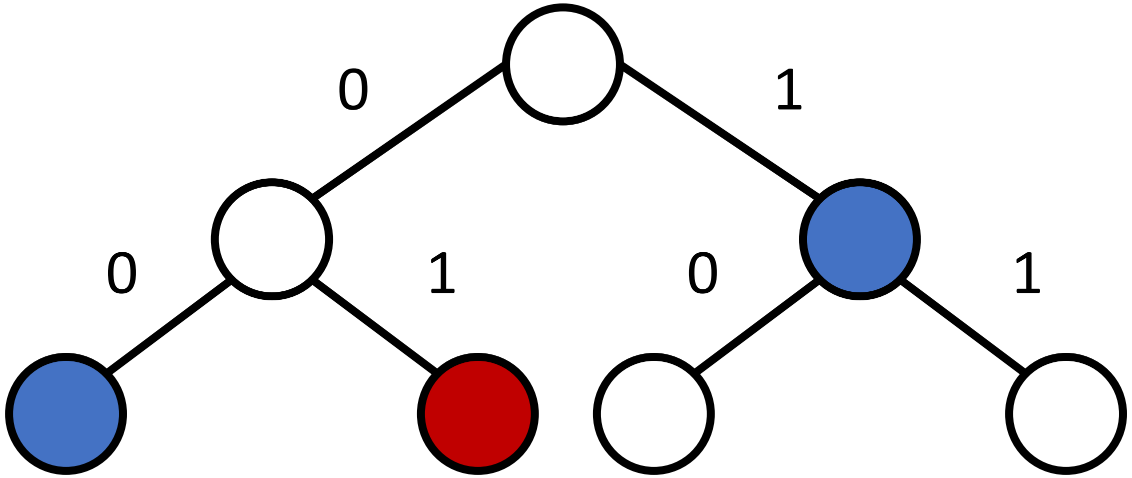
Combinatoric Approach to Revocation

[NNL01]

[NNL01]: Combinatoric approach for revocation based on subset-cover set systems



$\text{Encode}(x) \rightarrow \mathcal{J}_x$



$\text{ComputeCover}(\mathcal{L}) \rightarrow \mathcal{J}_{\mathcal{L}}$

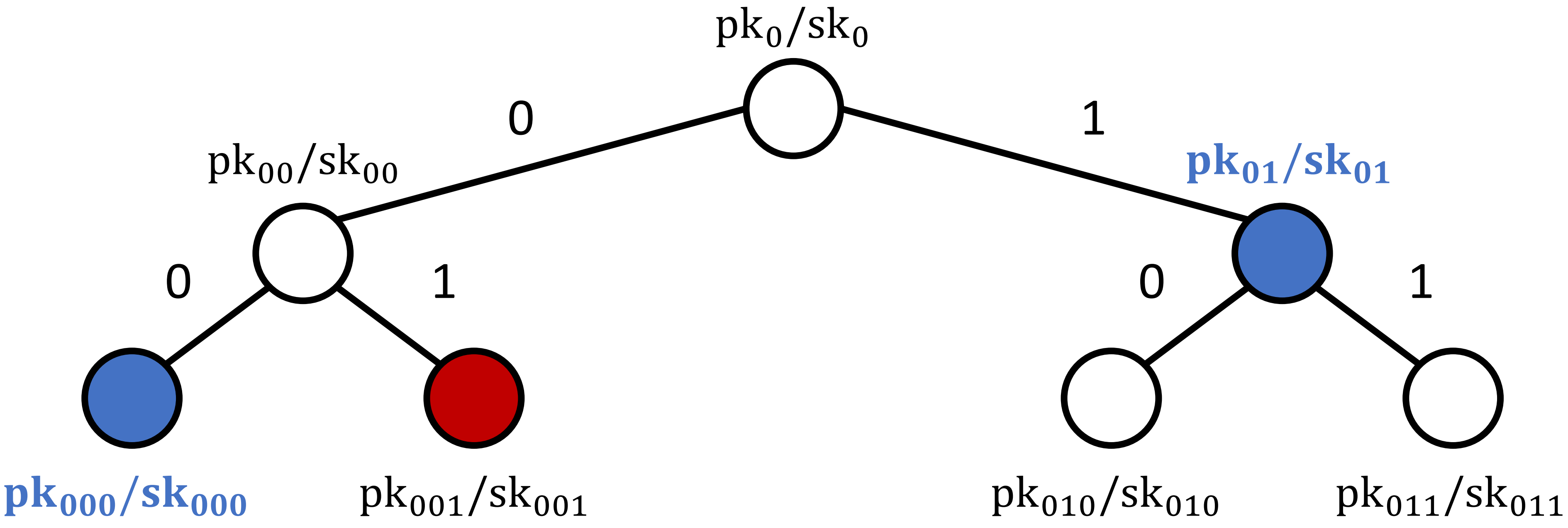
$\mathcal{J}_x$: Nodes associated with leaf x

$\mathcal{J}_{\mathcal{L}}$: Nodes that "cover" all leaves outside $\mathcal{L}$

Combinatoric Approach to Revocation

[NNL01]

[NNL01]: Combinatoric approach for revocation based on subset-cover set systems



Issue: number of public keys in this construction is <u>exponential</u>

Revocable Predicate Encryption

$\text{KeyGen}(\text{sk}, x)$

MFE secret key for x

Can decrypt ciphertexts ct_g where $g(x) = 1$

ABE key for function f

$f(\text{ct}_g) = \text{MFE.Decrypt}(\text{MFE.sk}_x, \text{ct}_g)$

Observation: ABE (or even IBE) can be used to “compress” the public keys into a short public parameters

Associate each key with an identity id

$\text{KeyGen}(\text{sk}, \text{id}, x)$

Can decrypt ciphertexts with attributes $(\text{id}^*, \text{ct}_g)$ where $\text{id} = \text{id}^*$ and $g(x) = 1$

ABE key for function f_{id}

$f_{\text{id}}(\text{id}^*, \text{ct}_g) =$

$\text{MFE.Decrypt}(\text{MFE.sk}_x, \text{ct}_g) \wedge (\text{id} = \text{id}^*)$

Revocable Predicate Encryption

$\mathrm{KeyGen}(\mathrm{sk}, x)$

MFE secret key for x

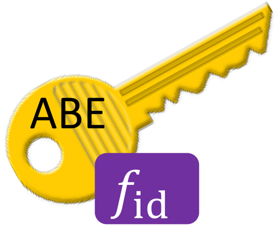
Can decrypt ciphertexts ct_g where $g(x) = 1$

ABE key for function f

$f(\mathrm{ct}_g) = \mathrm{MFE.Decrypt}(\mathrm{MFE.sk}_x, \mathrm{ct}_g)$

Observation: ABE (or even IBE) can be used to “compress” the public keys into a short public parameters

Can decrypt ciphertexts with attributes $(\mathrm{id}^*, \mathrm{ct}_g)$ where $\mathrm{id} = \mathrm{id}^*$ and $g(x) = 1$



ABE key for function f_{id}

$f_{\mathrm{id}}(\mathrm{id}^*, \mathrm{ct}_g) =$
$\mathrm{MFE.Decrypt}(\mathrm{MFE.sk}_x, \mathrm{ct}_g) \wedge (\mathrm{id} = \mathrm{id}^*)$

Revocation at ABE level ensures semantic security for revoked users (i.e., revoked keys cannot decrypt)

Revocable Predicate Encryption

$\mathrm{KeyGen}(\mathrm{sk}, x)$

MFE secret key for x

Approach does not extend to mixed FE (only supports public encryption to the all-ones function)

If we only have revocation for ABE keys, then scheme does <u>not</u> hide x (namely, can learn if $g(x) = 1$ even if $\mathrm{id} \neq \mathrm{id}^*$)

Can decrypt ciphertexts ct_g where $g(x) = 1$

ABE key for function f

$f(\mathrm{ct}_g) = \mathrm{MFE.Decrypt}(\mathrm{MFE.sk}_x, \mathrm{ct}_g)$

Can decrypt ciphertexts with attributes $(\mathrm{id}^*, \mathrm{ct}_g)$ where $\mathrm{id} = \mathrm{id}^*$ and $g(x) = 1$

ABE key for function f_{id}

$f_{\mathrm{id}}(\mathrm{id}^*, \mathrm{ct}_g) =$
$\mathrm{MFE.Decrypt}(\mathrm{MFE.sk}_x, \mathrm{ct}_g) \wedge (\mathrm{id} = \mathrm{id}^*)$

Revocable Predicate Encryption

$\text{KeyGen}(\text{sk}, x)$

MFE secret key for x

Can decrypt ciphertexts ct_g where $g(x) = 1$

ABE key for function f

$f(\text{ct}_g) = \text{MFE.Decrypt}(\text{MFE.sk}_x, \text{ct}_g)$

Can decrypt ciphertexts with attributes $(\text{id}^*, \text{ct}_g)$ where $\text{id} = \text{id}^*$ and $g(x) = 1$

Approach does not extend to mixed FE (only supports public encryption to the all-ones function)

If we only have revocation for ABE keys, then scheme does not hide x (namely, can learn if $g(x) = 1$ even if $\text{id} \neq \text{id}^*$)

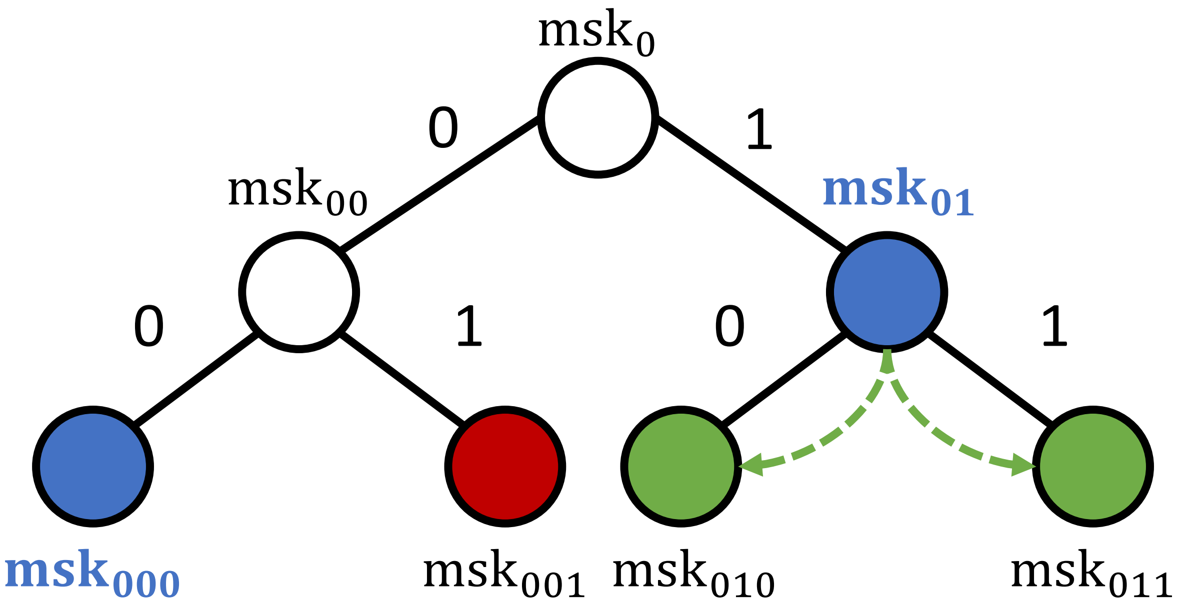
Does not satisfy (strong) attribute-hiding: **problematic for tracing**

Revocable Predicate Encryption

$\mathsf{KeyGen}(\mathsf{sk}, x)$

MFE secret key for x

Derive msks from a PRF:
$\mathsf{msk}_i \leftarrow \mathsf{PRF}(k, i)$



Can decrypt ciphertexts with attributes $(\mathsf{id}^*, \mathsf{ct}_g)$ where $\mathsf{id} = \mathsf{id}^*$ and $g(x) = 1$

ABE key for function f_{id}

$f_{\mathsf{id}}(\mathsf{id}^*, \mathsf{ct}_g) =$
$\mathsf{MFE.Decrypt}(\mathsf{MFE.sk}_x, \mathsf{ct}_g) \wedge (\mathsf{id} = \mathsf{id}^*)$

Observation: master secret key in existing mixed FE schemes can be sampled *after* the public parameters

All master secret keys in the tree share a *common* set of public parameters pp

Putting the Pieces Together

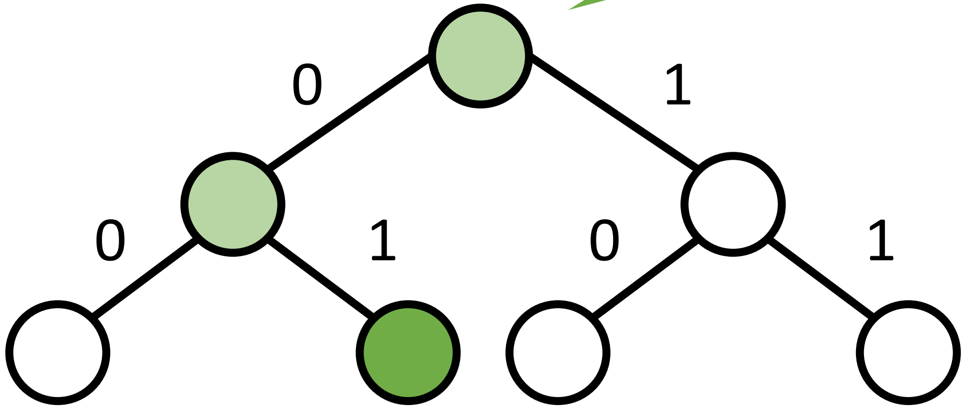
Public parameters: mpk (for ABE scheme) and pp (for mixed FE scheme)

Master secret key: msk (for ABE scheme) and k (for PRF)

Putting the Pieces Together

KeyGen(msk, id, x)

Step 1: $\text{Encode}(\text{id}) \to \mathcal{J}_{\text{id}}$



Step 2: For each node i in $\mathcal{J}_{\text{id}}$:

- Sample MFE master secret key:
  $\text{MFE.msk}_i \leftarrow \text{MFE.MSKGen}(\text{MFE.pp}; \text{PRF}(k, i))$
- Issue MFE secret key for x:
  $\text{MFE.sk}_{i,x} \leftarrow \text{MFE.KeyGen}(\text{MFE.msk}_i, x)$
- Issue ABE secret key (MFE key + id hard-wired)
  $\text{ABE.sk}_{i,x} \leftarrow \text{ABE.KeyGen}(\text{ABE.msk}, f_{\text{id}})$

MFE secret key for x
(with respect to node i)

$f_{\text{id}}(\text{id}^*, \text{ct}_g) = 1$ if:

- $\text{MFE.Decrypt}(\text{MFE.sk}_{i,x}, \text{ct}_g)$
- $\text{id} = \text{id}^*$

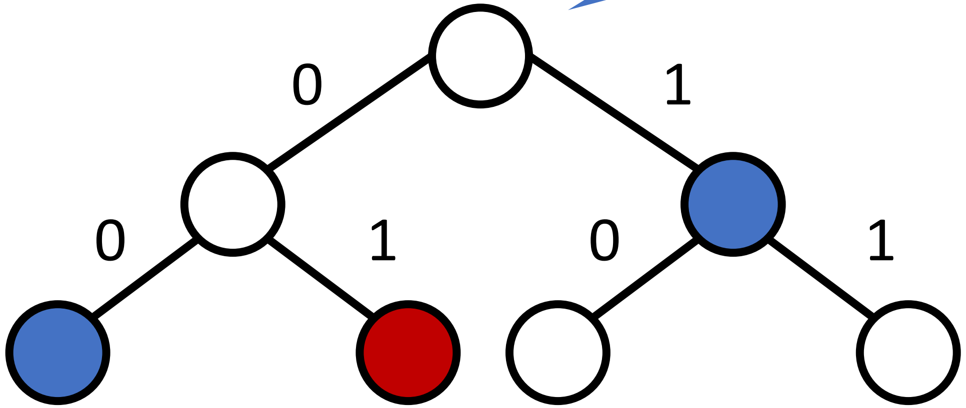
Putting the Pieces Together

Broadcast$(\mathbf{pk}, \boldsymbol{m}, \boldsymbol{\mathcal{L}})$

Step 1: $\text{ComputeCover}(\mathcal{L}) \rightarrow \mathcal{J}_{\mathcal{L}}$

Step 2: For each node i in $\mathcal{J}_{\mathcal{L}}$:

- Sample MFE ciphertext
  $$\text{MFE.ct}_i \leftarrow \text{MFE.PKEnc}(\text{MFE.pp})$$
- Encrypt message using ABE
  $$\text{ABE.ct}_i \leftarrow \text{ABE.Enc}(\text{ABE.pp}, (\text{MFE.ct}_i, i), m)$$



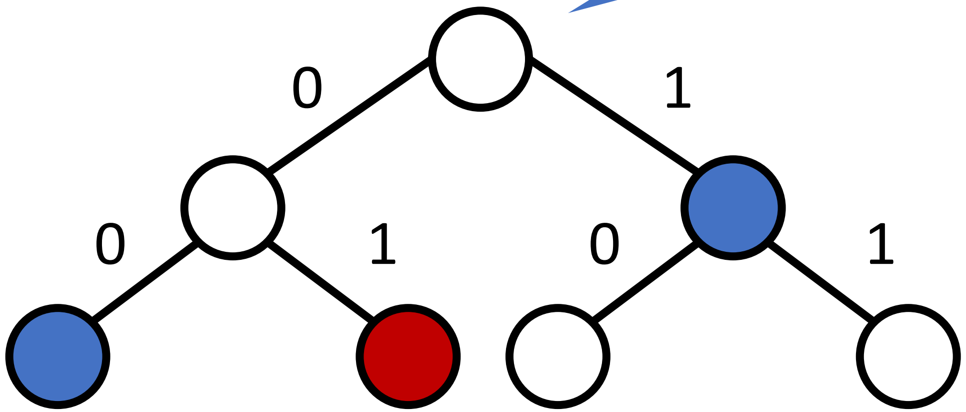
public MFE ciphertext
(for all-ones function)

ABE ciphertext

Putting the Pieces Together

$\mathbf{Encrypt}(\mathbf{msk}, \boldsymbol{g}, \boldsymbol{m}, \boldsymbol{\mathcal{L}})$

Step 1: $\text{ComputeCover}(\mathcal{L}) \to \mathcal{J}_{\mathcal{L}}$



Step 2: For each node i in $\mathcal{J}_{\mathcal{L}}$:

- Sample MFE master secret key:
  $\text{MFE.msk}_i \leftarrow \text{MFE.MSKGen}\big(\text{MFE.pp}; \text{PRF}(k, i)\big)$
- Sample MFE ciphertext
  $\text{MFE.ct}_i \leftarrow \text{MFE.SKEnc}(\text{MFE.msk}_i, g)$
- Encrypt message using ABE
  $\text{ABE.ct}_i \leftarrow \text{ABE.Enc}(\text{ABE.pp}, (\text{MFE.ct}_i, i), m)$

public MFE ciphertext (for function g)

ABE ciphertext

Putting the Pieces Together

Assuming sub-exponential hardness of LWE, there exists a
fully collusion-resistant identity-based trace-and-revoke scheme

$|\mathrm{sk}| = n \cdot \mathrm{poly}(\lambda, \log n)$

$|\mathrm{ct}| = |m| + |\mathcal{L}| \cdot \mathrm{poly}(\lambda, \log n)$

Encryption algorithm is public-key

Tracing algorithm is secret-key

m: message

n: bit-length of identity

$\mathcal{L}$: revocation list

Open Problems

Assuming sub-exponential hardness of LWE, there exists a fully collusion-resistant identity-based trace-and-revoke scheme

$|\mathrm{sk}| = n \cdot \mathrm{poly}(\lambda, \log n)$

$|\mathrm{ct}| = |m| + |\mathcal{L}| \cdot \mathrm{poly}(\lambda, \log n)$

Encryption algorithm is public-key

Tracing algorithm is secret-key

- Succinct broadcast: Ciphertext size scaling sublinearly in the number of revoked users (i.e., description length of $\mathcal{L}$)
- Support public tracing
- Polynomial hardness (polynomial hardness of LWE suffices for identity-based traitor tracing [GKW19])

Thank you!

https://eprint.iacr.org/2019/984