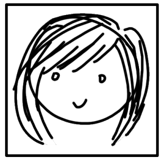
Public Key Function-Private Hidden Vector Encryption (and More)

James Bartusek	(UC Berkeley)
Brent Carmer	(Galois)
Abhishek Jain	(Johns Hopkins)
Zhengzhong Jin	(Johns Hopkins)
Tancrède Lepoint	(Google)
Fermi Ma	(Princeton & NTT Research)
Tal Malkin	(Columbia)
Alex J. Malozemoff	(Galois)
Mariana Raykova	(Google)

Hey Alice,
It’s me, Bob.



Alice’s homepage
alice@gmail.com

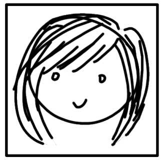
my public key is:
8h9f8he9
ak928ads

Enc(8h9f8he9 ak928ads , Hey Alice, It's me, Bob.)

pk

↓

ct

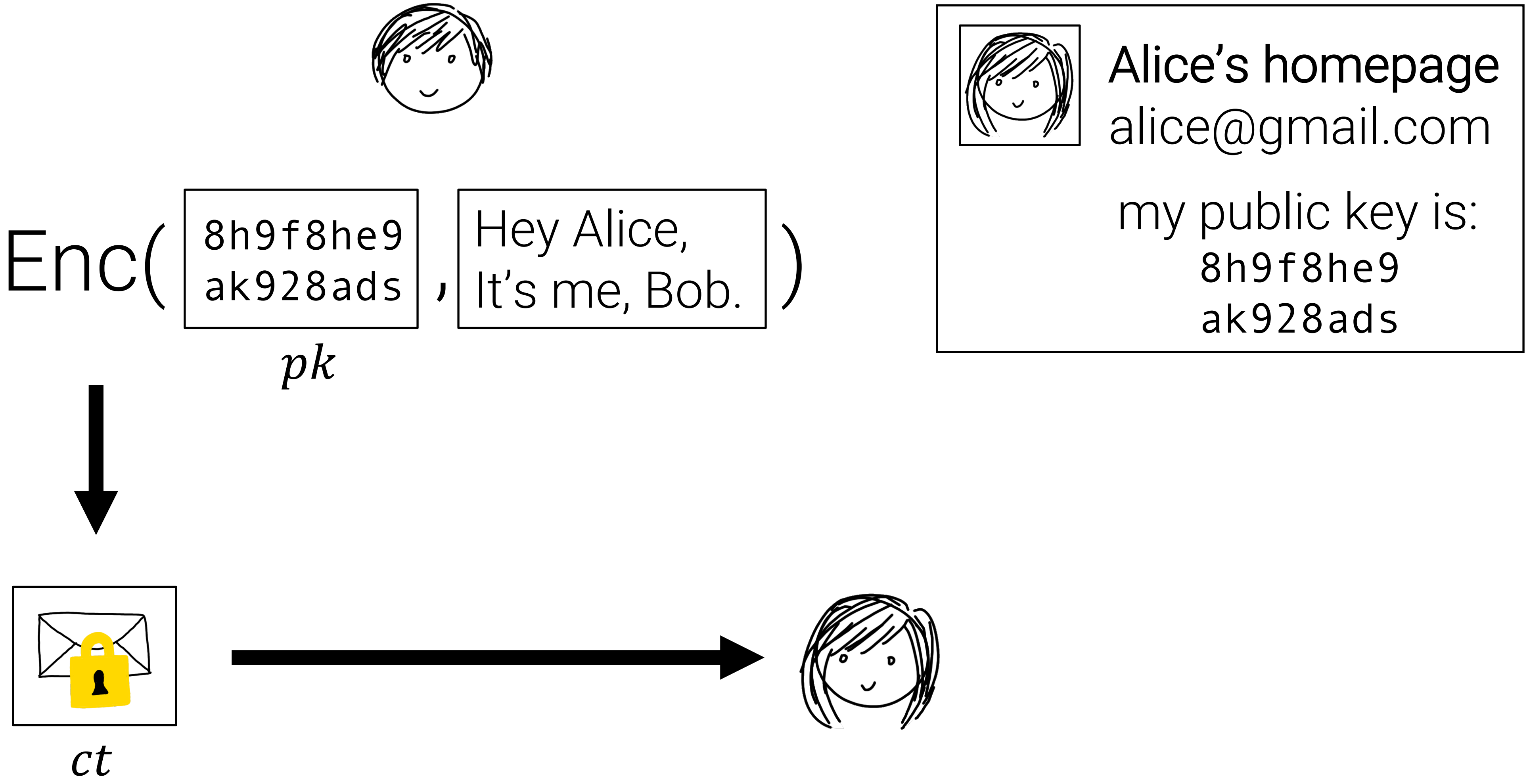


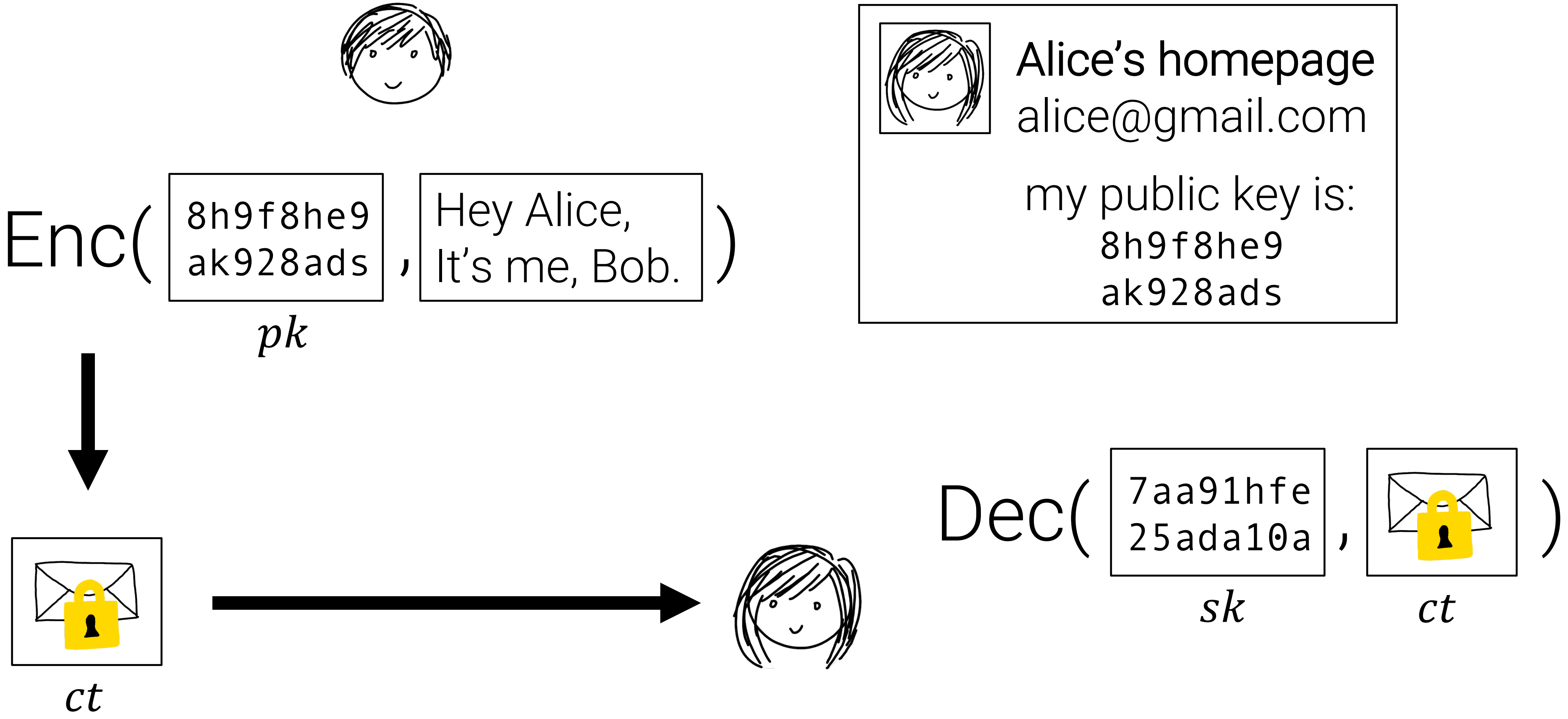
Alice's homepage

alice@gmail.com

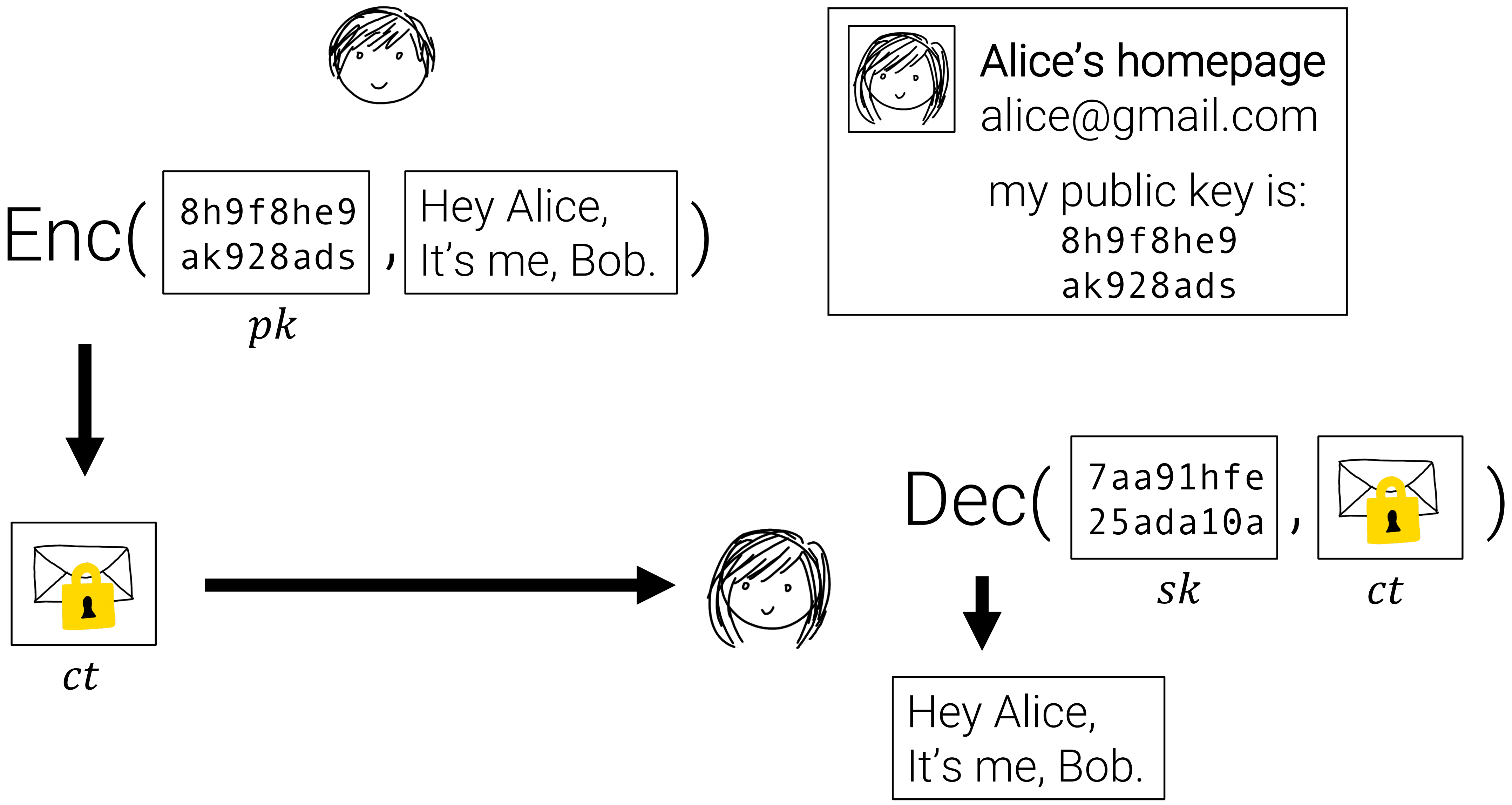
my public key is:

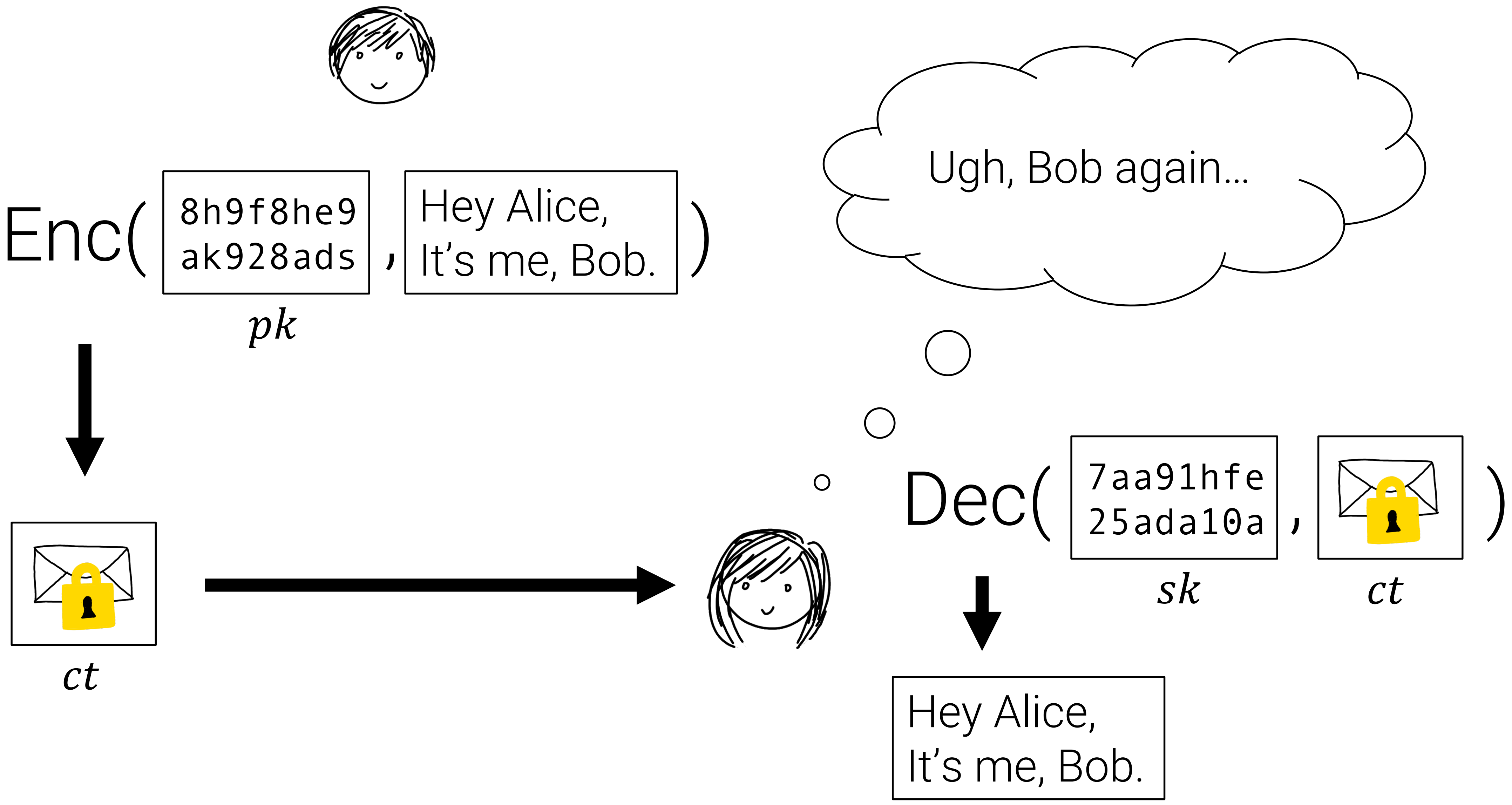
8h9f8he9
ak928ads

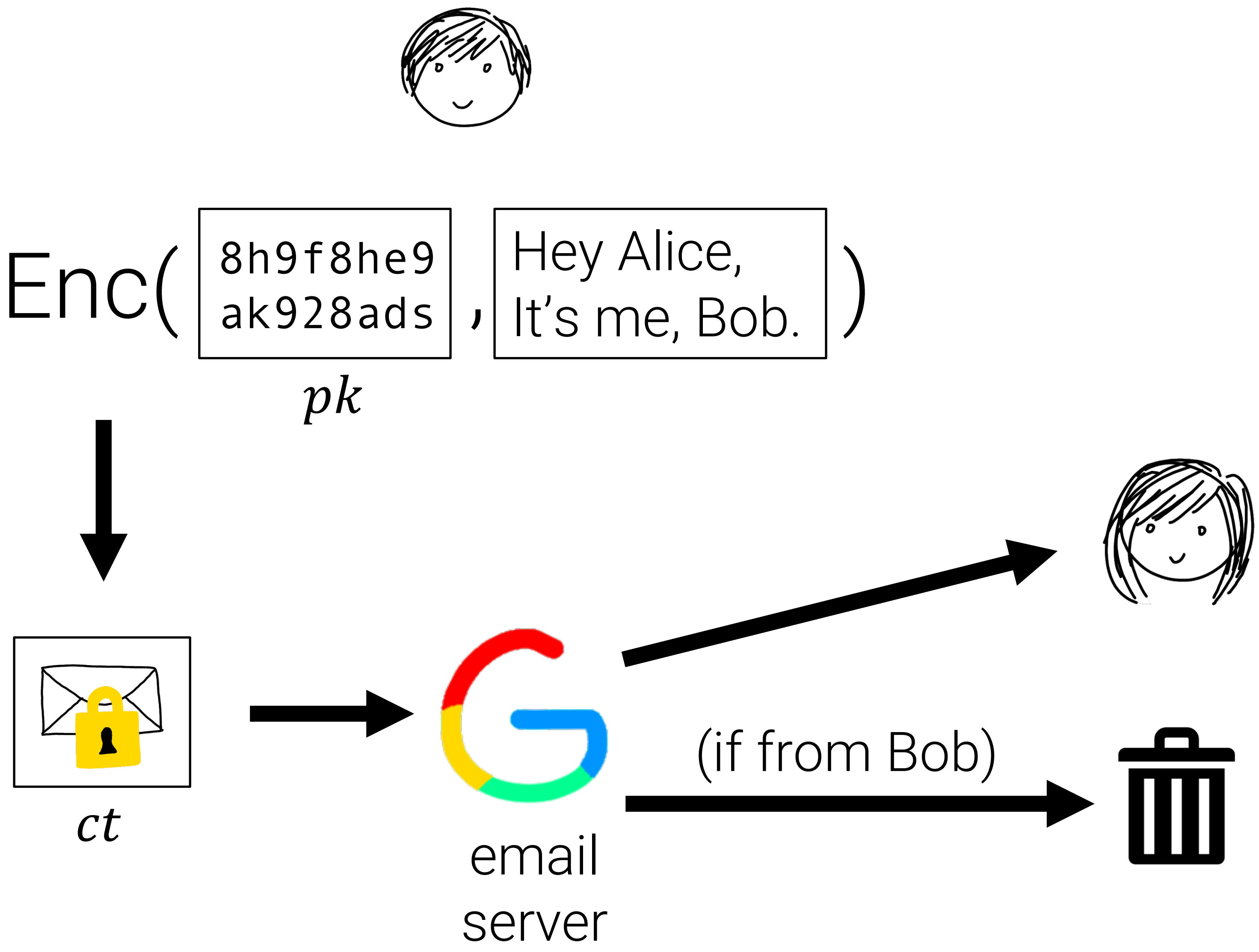

Enc(
8h9f8he9
ak928ads
Hey Alice,
It's me, Bob.
)
pk
ct
Alice's homepage
alice@gmail.com
my public key is:
8h9f8he9
ak928ads



Enc(8h9f8he9 ak928ads , Hey Alice, It's me, Bob.)
pk
ct
Alice's homepage
alice@gmail.com
my public key is:
8h9f8he9
ak928ads
Dec(7aa91hfe 25ada10a ,)
sk
ct


Alice's homepage
alice@gmail.com
my public key is:
8h9f8he9
ak928ads
Enc(
8h9f8he9
ak928ads
,
Hey Alice,
It's me, Bob.
)
pk
ct
Dec(
7aa91hfe
25ada10a
,
)
sk
ct
Hey Alice,
It's me, Bob.

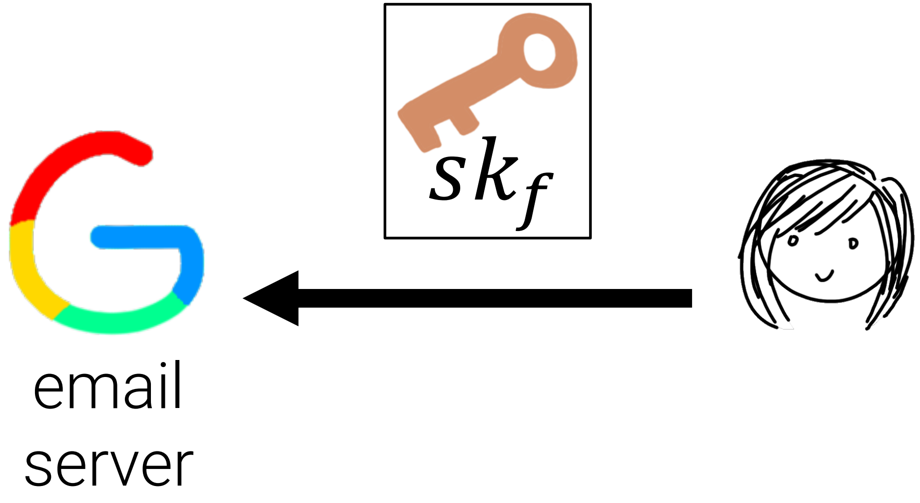

Enc(
8h9f8he9
ak928ads
pk
Hey Alice,
It's me, Bob.
)
ct
Ugh, Bob again...
Dec(
7aa91hfe
25ada10a
sk
ct
)
Hey Alice,
It's me, Bob.



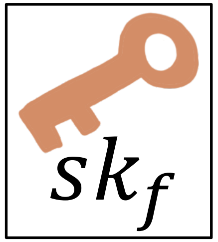
Goal: Allow G to filter emails, without sacrificing privacy

Predicate Encryption

[BCOP04,SW05,BW07,KSW08]



$$f(x) = \begin{cases} 1 & \text{if } x \text{ contains "It's me, Bob"} \\ 0 & \text{else} \end{cases}$$

Given 

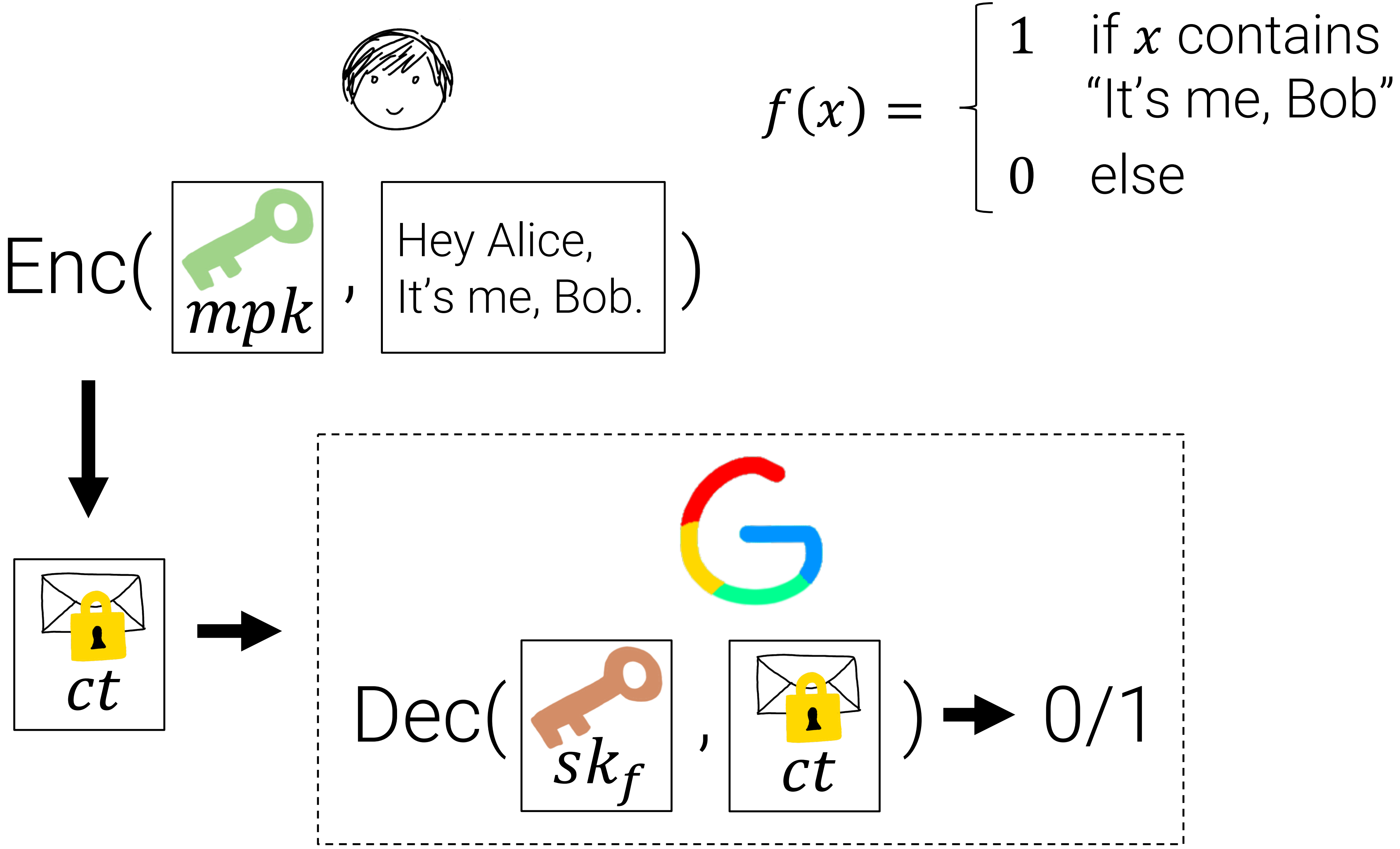
, can learn $f(x)$ given encryption of x

Security: Nothing else about x is leaked

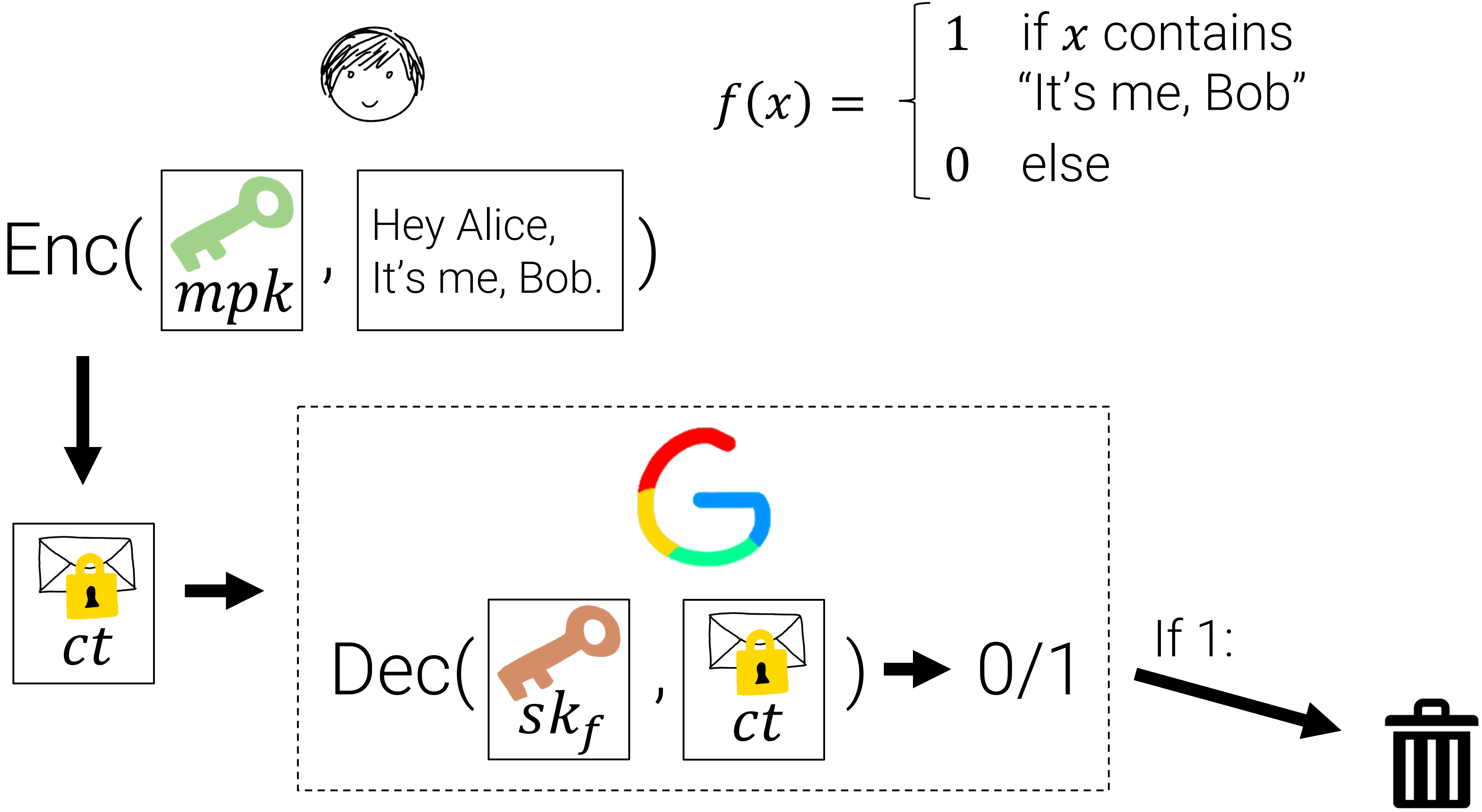
$$f(x) = \begin{cases} 1 & \text{if } x \text{ contains "It's me, Bob"} \\ 0 & \text{else} \end{cases}$$

Enc( mpk , Hey Alice, It's me, Bob.)

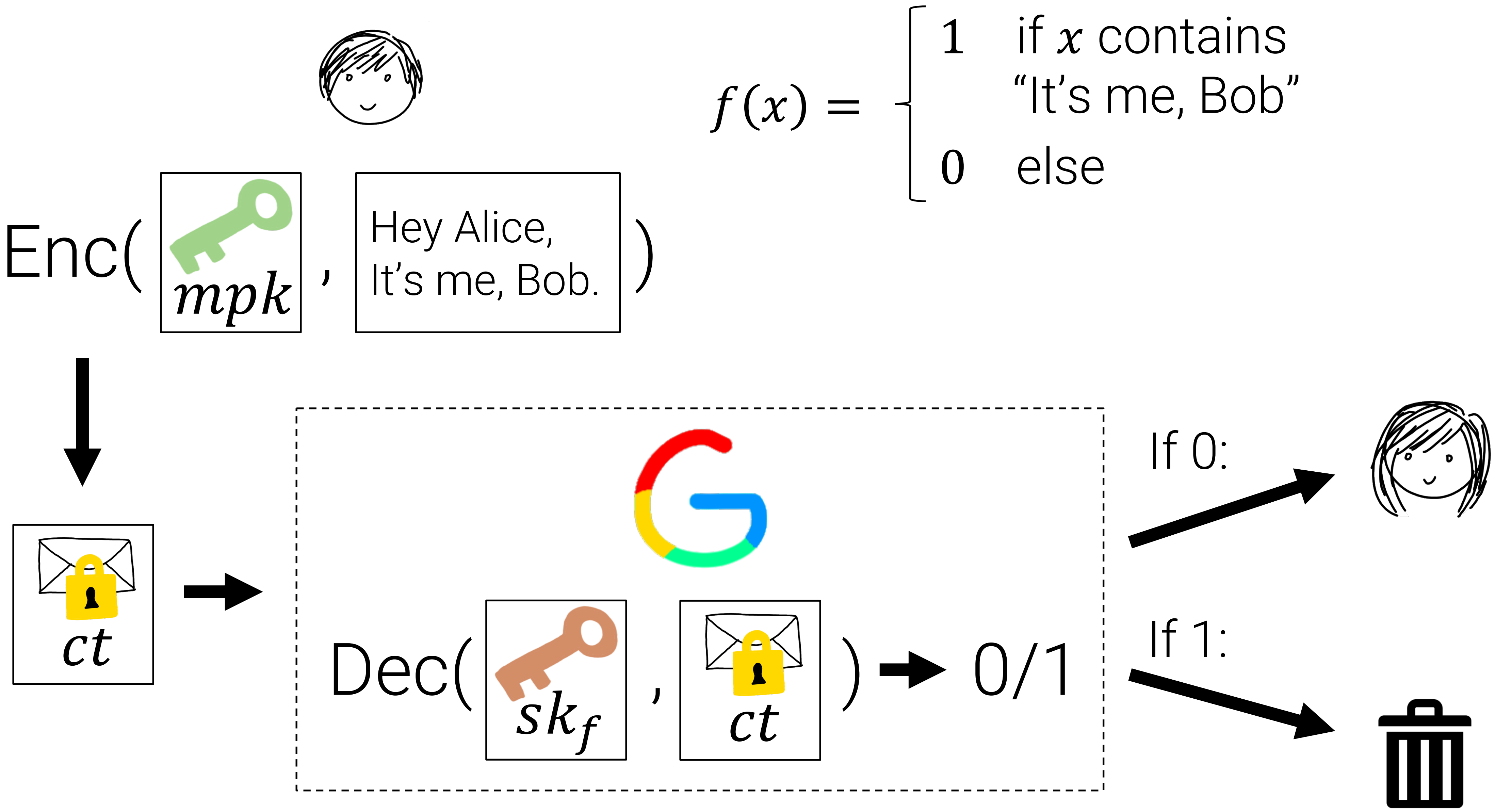
ct



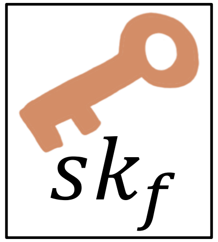
$f(x) = \begin{cases} 1 & \text{if } x \text{ contains "It's me, Bob"} \\ 0 & \text{else} \end{cases}$
Enc(mpk , Hey Alice, It's me, Bob.)
ct
Dec(sk_f , ct) → 0/1



$f(x) = \begin{cases} 1 & \text{if } x \text{ contains "It's me, Bob"} \\ 0 & \text{else} \end{cases}$
Enc(mpk , Hey Alice, It's me, Bob.)
ct
Dec(sk_f , ct) → 0/1
If 1:



$f(x) = \begin{cases} 1 & \text{if } x \text{ contains "It's me, Bob"} \\ 0 & \text{else} \end{cases}$
Enc(
mpk
Hey Alice,
It's me, Bob.
)
ct
Dec(
sk_f
ct
) → 0/1
If 0:
If 1:

In many schemes, 

does not hide f

In many schemes,  sk_f does not hide f

sk_f

$$f(x) = \begin{cases} 1 & \text{if } x \text{ contains "It's me, Bob"} \\ 0 & \text{else} \end{cases}$$

In many schemes,  sk_f does not hide f

sk_f

$$f(x) = \begin{cases} 1 & \text{if } x \text{ contains “It's me, Bob”} \\ 0 & \text{else} \end{cases}$$

In many schemes,  sk_f does not hide f

sk_f

Function Privacy

[SWP00,OS07,BSW09,SSW09]

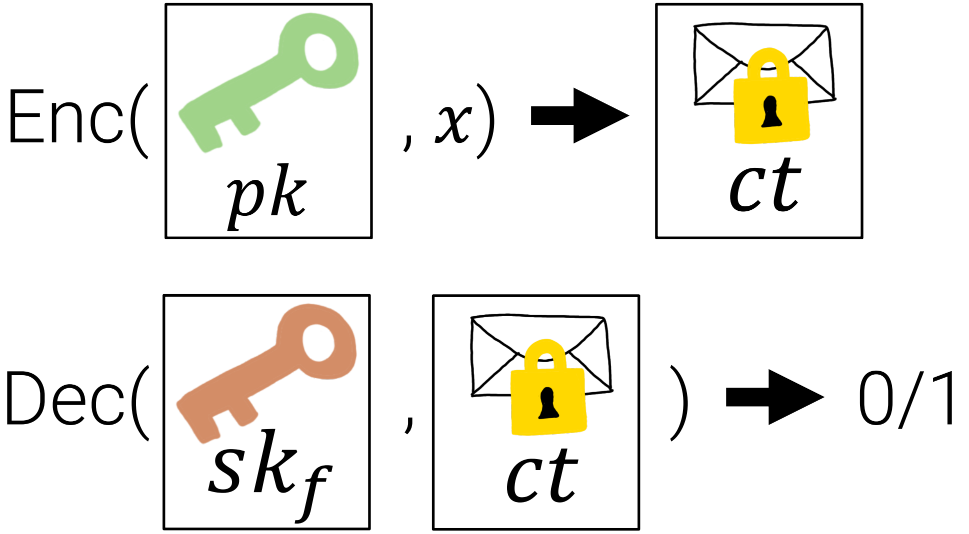
What does function privacy mean in the public-key setting?

What does function privacy mean in the public-key setting?

Potential issue: can evaluate $f(x)$ for any x it wants

What does function privacy mean in the public-key setting?

Potential issue:  can evaluate $f(x)$ for any x it wants



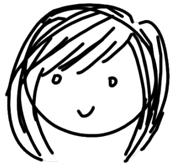
What does function privacy mean in the public-key setting?

Restriction: f sampled s.t. hard to find x where $f(x) = 1$.

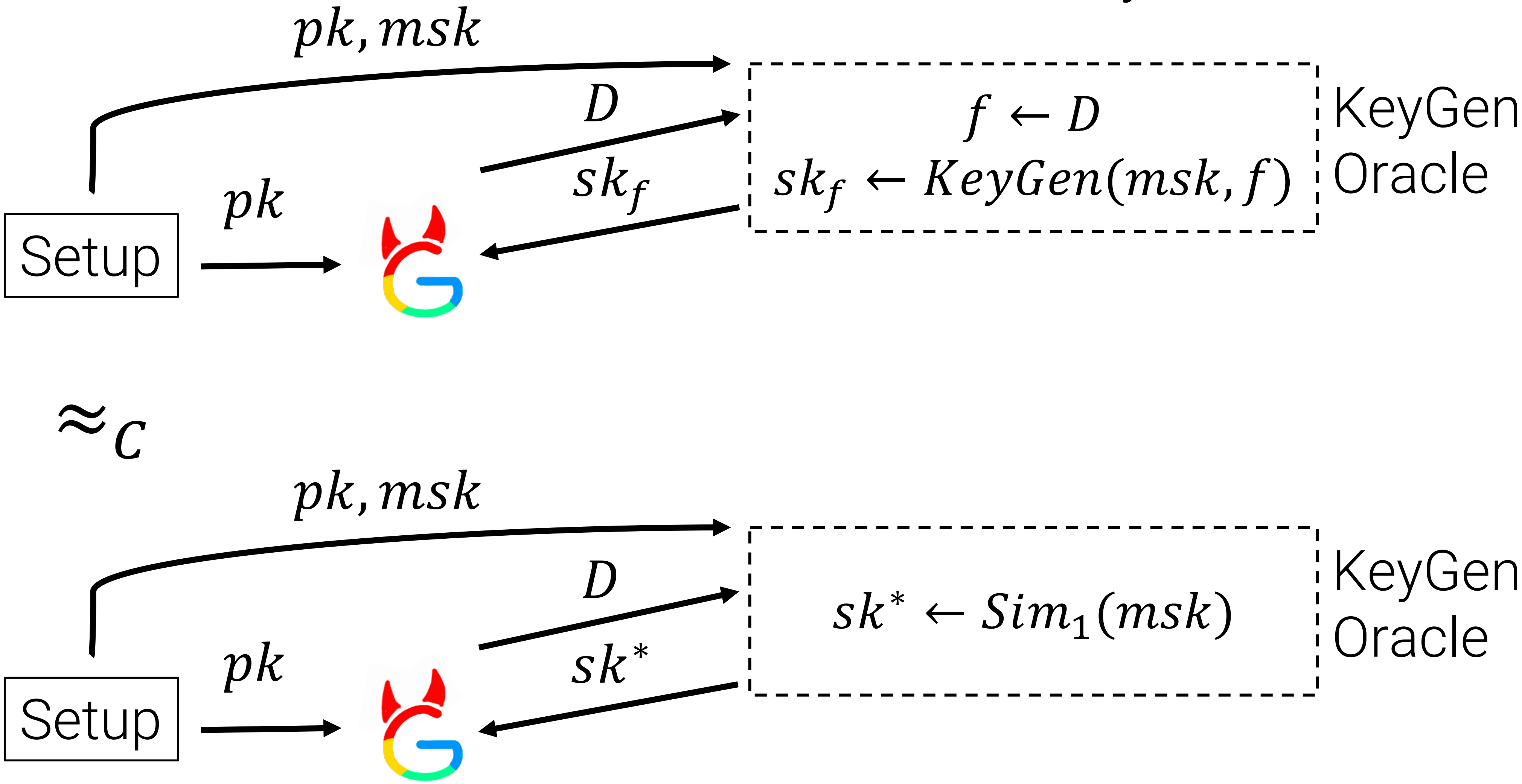
What does function privacy mean in the public-key setting?

Restriction: f sampled s.t. hard to find x where $f(x) = 1$.

Why is this reasonable?

Intuition:  unlikely to blindly guess what  is filtering for (i.e. will never find an x s.t. $f(x) = 1$).

Function Privacy [BRS13a]

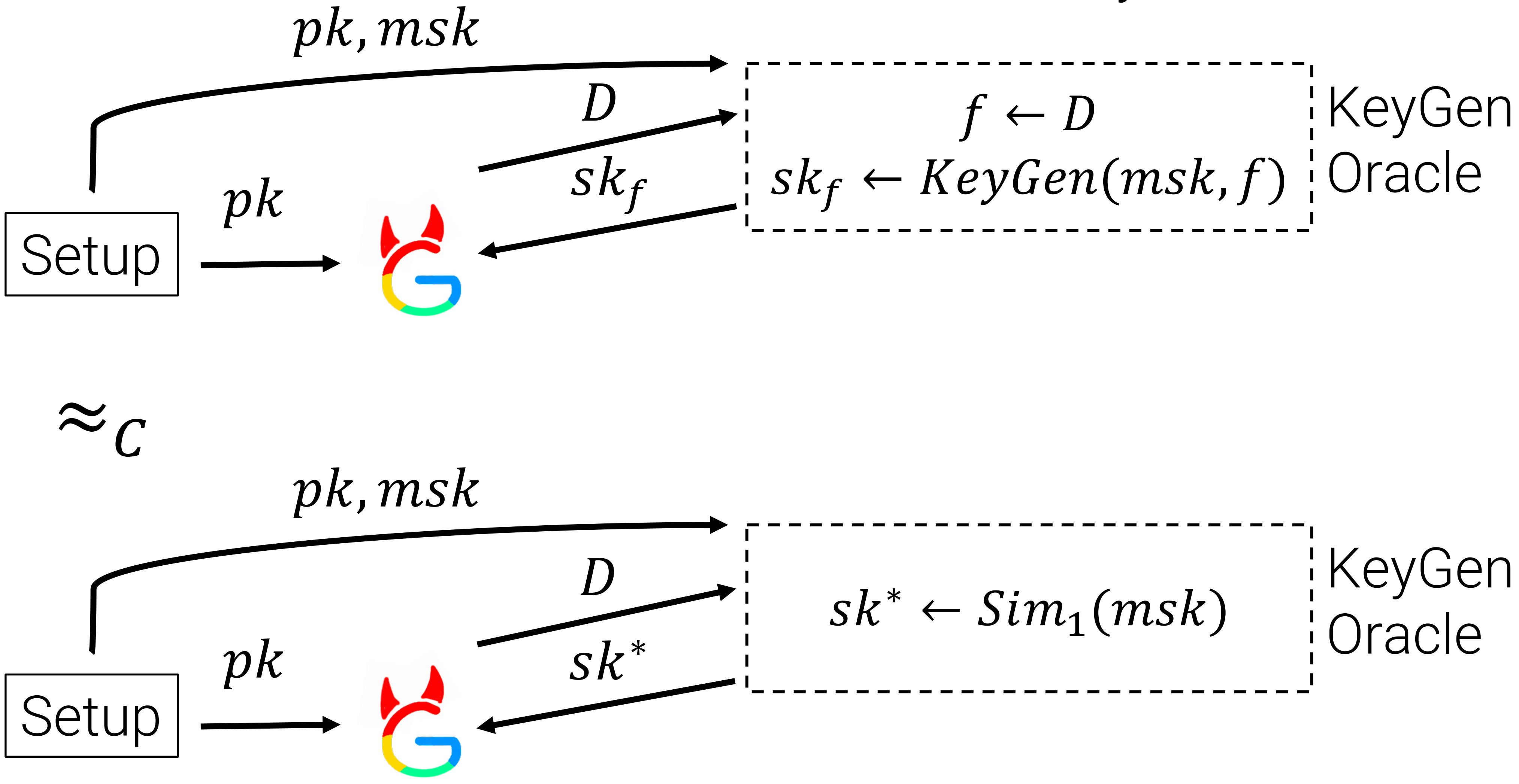


But in reality, may receive encryptions of x such that $f(x) = 1$; it just can't generate these encryptions for itself

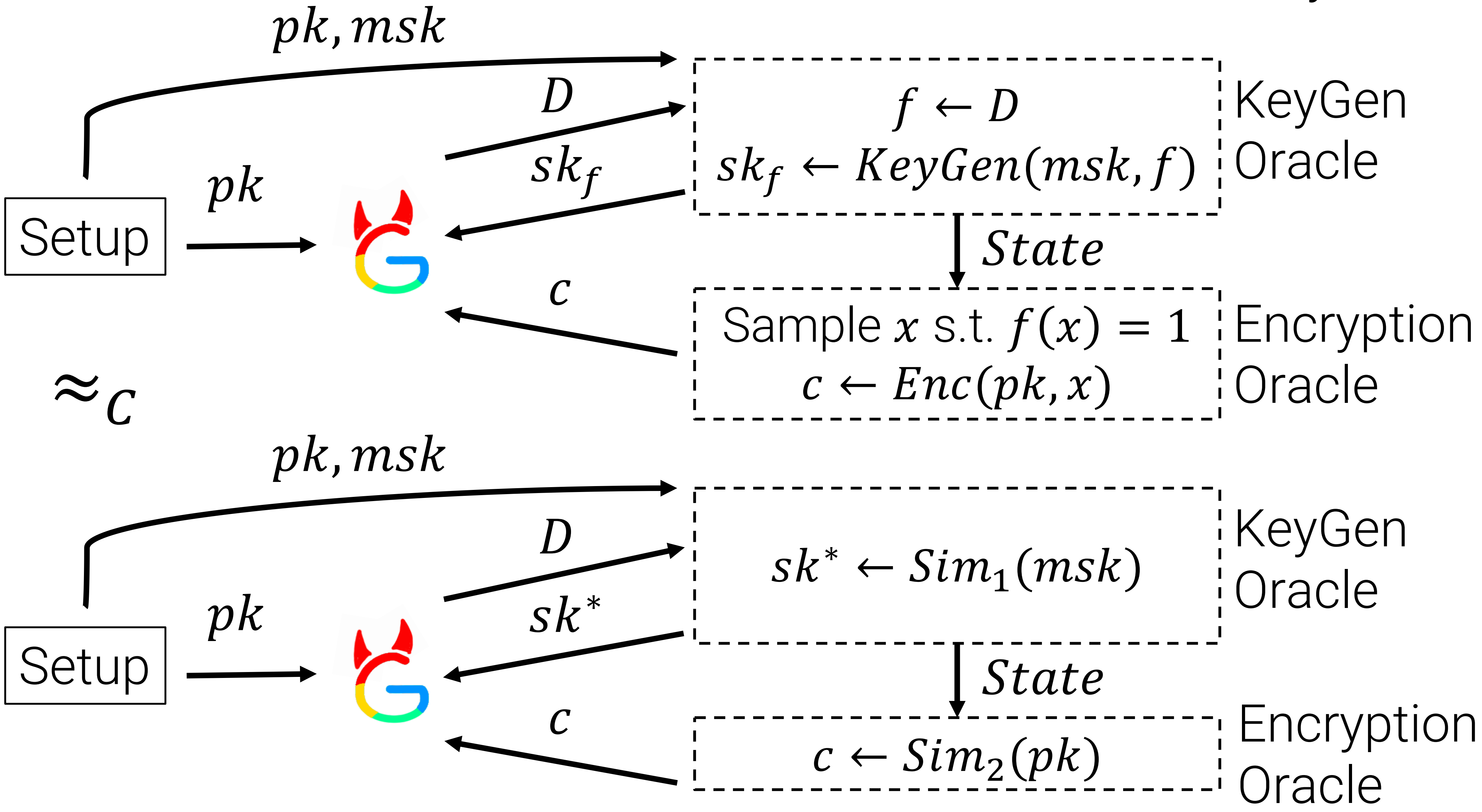
But in reality, [G] may receive encryptions of x such that $f(x) = 1$; it just can't generate these encryptions for itself

[BRS13a] address this with **enhanced function privacy**, where an "encryption oracle" is provided

Function Privacy [BRS13a]



“Enhanced” Function Privacy [BRS13a]



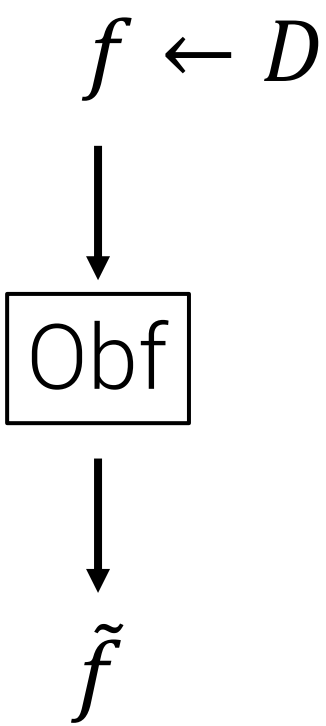
	Predicates	Assumption	Enhanced Function Privacy?
[BRS13a]	• Equality (IBE)	None (statistical)	Yes
[BRS13b]	• Subspace Membership	None (statistical)	No
[PMR19]	• Conjunctions** (Hidden Vector Encryption)	Strong Matrix DDH	No

**leaks positions of "wildcards"

	Predicates	Assumption	Enhanced Function Privacy?
[BRS13a]	• Equality (IBE)	None (statistical)	Yes
[BRS13b]	• Subspace Membership	None (statistical)	No
[PMR19]	• Conjunctions** (Hidden Vector Encryption)	Strong Matrix DDH	No
This work	• Equality (IBE) • Conjunctions (Hidden Vector Encryption) • “Small Superset” [BW19]	Generic Group Model	Yes

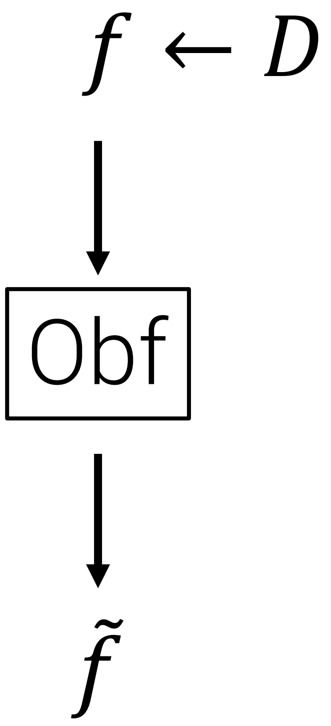
**leaks positions of “wildcards”

Our Techniques: Obfuscation in the Distributional Setting



1) Functionality: Given $\tilde{f}$, possible to learn $f(x)$ for any x
2) Distributional Security: If D is high-entropy, then $\tilde{f}$ leaks **nothing** about f

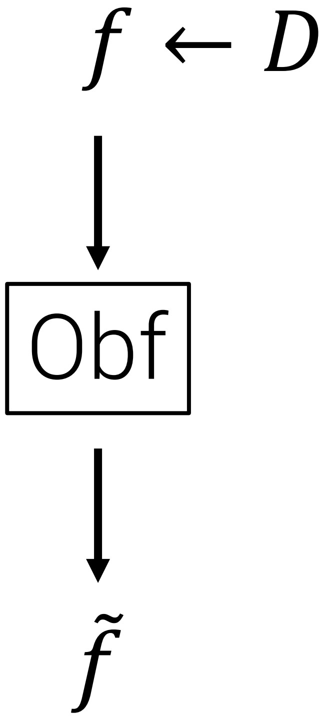
Our Techniques: Obfuscation in the Distributional Setting



1) Functionality: Given $\tilde{f}$, possible to learn $f(x)$ for any x
2) Distributional Security: If D is high-entropy, then $\tilde{f}$ leaks **nothing** about f

Point Functions: $f_x(y) = 1$ iff $y = x$.
Sample x with high entropy.
(x, y length n bitstrings)

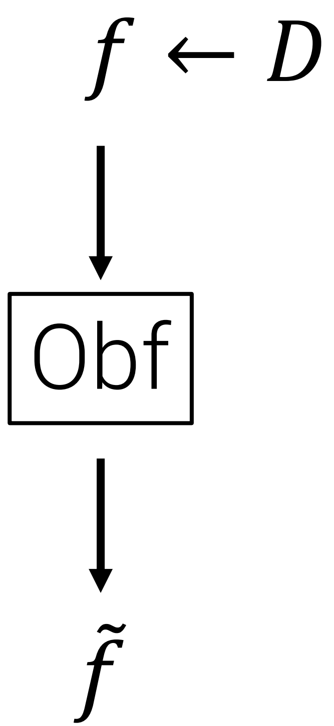
Our Techniques: Obfuscation in the Distributional Setting



1) Functionality: Given $\tilde{f}$, possible to learn $f(x)$ for any x
2) Distributional Security: If D is high-entropy, then $\tilde{f}$ leaks **nothing** about f

Conjunctions: $f_{pat}(\cdot)$. $pat =$ 0*1**10.
$f_{pat}(y) = 1$ if y agrees with pat on all 0/1 positions (y is length n bitstring)

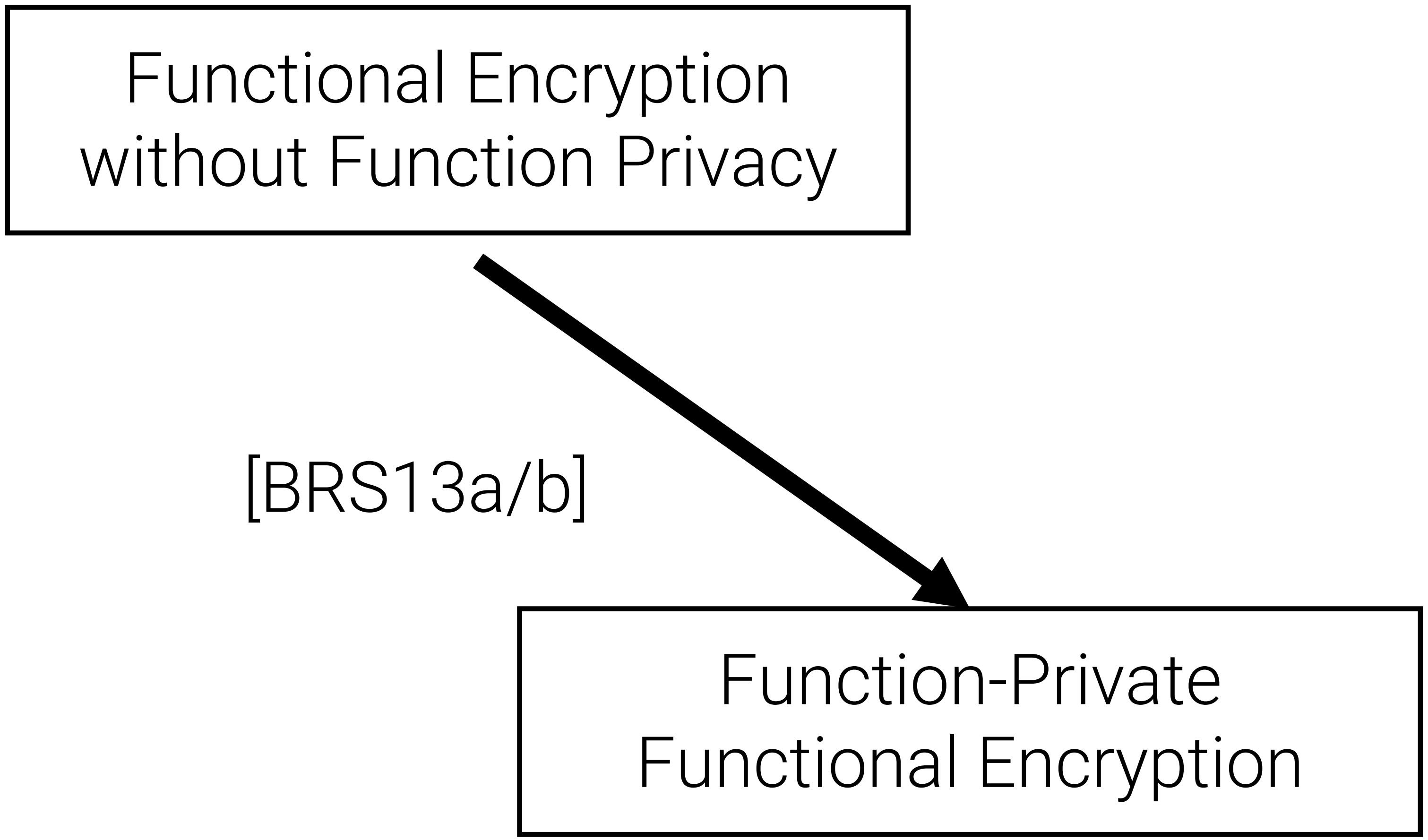
Our Techniques: Obfuscation in the Distributional Setting

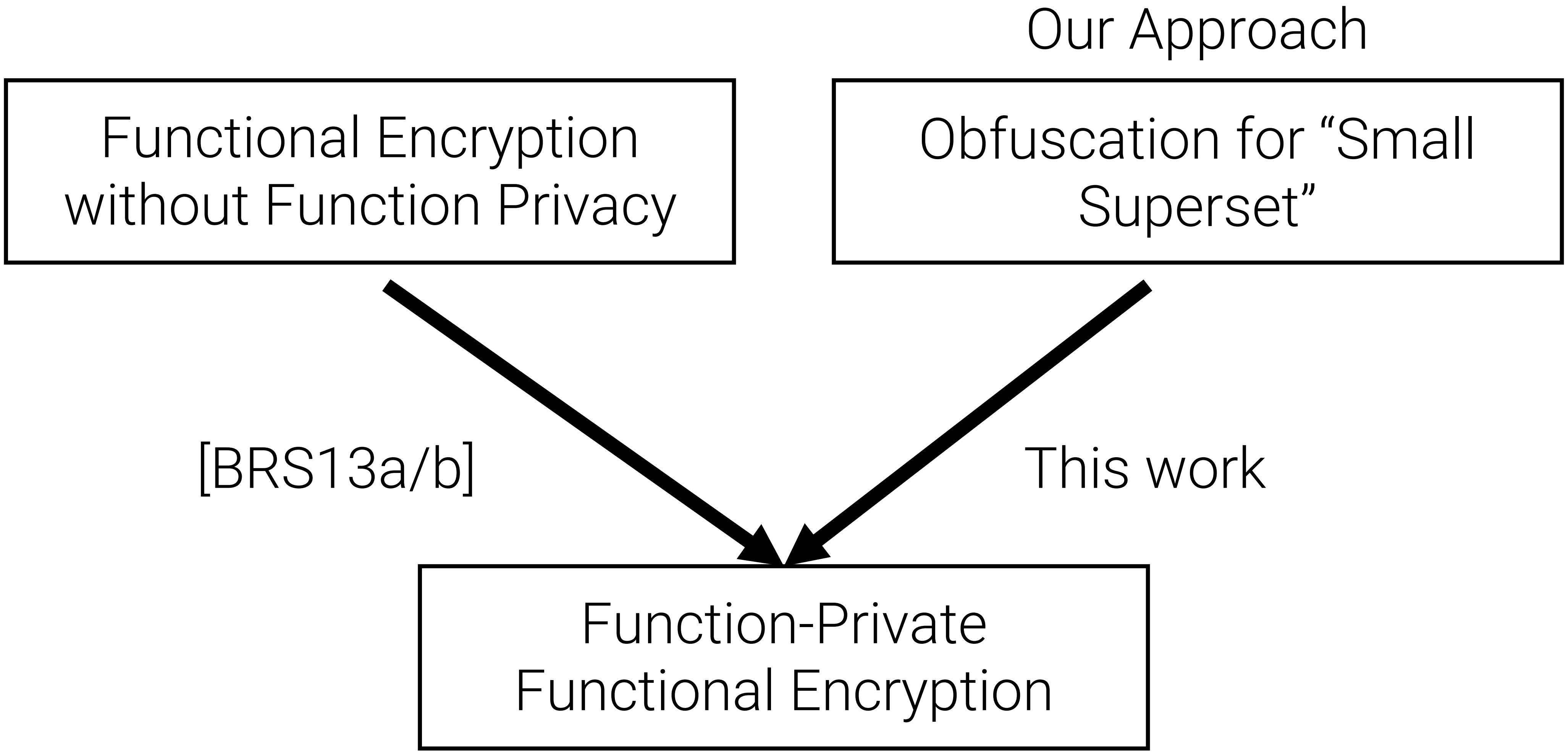


1) Functionality: Given $\tilde{f}$, possible to learn $f(x)$ for any x
2) Distributional Security: If D is high-entropy, then $\tilde{f}$ leaks **nothing** about f

Observation: Think of $\tilde{f}$ as sk_f (a function decryption key) which is evaluated on "trivially insecure" ciphertexts

Can we “upgrade” specific obfuscation constructions so that they can be evaluated on **encrypted** inputs?


Functional Encryption without Function Privacy
[BRS13a/b]
Function-Private Functional Encryption


Our Approach
Functional Encryption without Function Privacy
Obfuscation for "Small Superset"
[BRS13a/b]
This work
Function-Private Functional Encryption

“Small Superset” Obfuscation [BKMPRS18, BLMZ19, BW19]

For subset $X \subseteq [n]$, define $f_{t,X}(S) = \begin{cases} 1 \text{ if } X \subseteq S \text{ AND } |S| \leq t. \\ 0 \text{ otherwise} \end{cases}$

(Generalizes point functions, conjunctions, and more)

“Small Superset” Obfuscation [BKMPRS18, BLMZ19, BW19]

For subset $X \subseteq [n]$, define $f_{t,X}(S) = \begin{cases} 1 \text{ if } X \subseteq S \text{ AND } |S| \leq t. \\ 0 \text{ otherwise} \end{cases}$

Obfuscation for $t = 4, X = \{1,5,11\}$

1) Sample random $\vec{v}$ in rowspace of $\underbrace{\begin{bmatrix} 1 & 1^2 & 1^3 & 1^4 & 1^5 \\ 5 & 5^2 & 5^3 & 5^4 & 5^5 \\ 11 & 11^2 & 11^3 & 11^4 & 11^5 \end{bmatrix}}_{t+1}$
2) Output $g^{\vec{v}}$.

"Small Superset" Obfuscation [BKMPRS18, BLMZ19, BW19]

For subset $X \subseteq [n]$, define $f_{t,X}(S) = \begin{cases} 1 \text{ if } X \subseteq S \text{ AND } |S| \leq t. \\ 0 \text{ otherwise} \end{cases}$

Obfuscation for $t = 4, X = \{1,5,11\}$

1) Sample random $\vec{v}$ in rowspace of $\underbrace{\begin{bmatrix} 1 & 1^2 & 1^3 & 1^4 & 1^5 \\ 5 & 5^2 & 5^3 & 5^4 & 5^5 \\ 11 & 11^2 & 11^3 & 11^4 & 11^5 \end{bmatrix}}_{t+1}$
2) Output $g^{\vec{v}}$.

Evaluate on $Y = \{1,5,6,11\}$

given obfuscation $(g^{v_1}, \ldots, g^{v_{t+1}})$

1) Let $\vec{w}$ be random in kernel of $\underbrace{\begin{bmatrix} 1 & 1^2 & 1^3 & 1^4 & 1^5 \\ 5 & 5^2 & 5^3 & 5^4 & 5^5 \\ 6 & 6^2 & 6^3 & 6^4 & 6^5 \\ 11 & 11^2 & 11^3 & 11^4 & 11^5 \end{bmatrix}}_{t+1}$
2) Output 1 iff $g^{\vec{v} \cdot \vec{w}} = g^0$.

"Small Superset" Obfuscation [BKMPRS18, BLMZ19, BW19]

For subset $X \subseteq [n]$, define $f_{t,X}(S) = \begin{cases} 1 \text{ if } X \subseteq S \text{ AND } |S| \leq t. \\ 0 \text{ otherwise} \end{cases}$

Obfuscation for $t = 4, X = \{1,5,11\}$

1) Sample random $\vec{v}$ in rowspace of

$$M_{t,X} = \underbrace{\begin{bmatrix} 1 & 1^2 & 1^3 & 1^4 & 1^5 \\ 5 & 5^2 & 5^3 & 5^4 & 5^5 \\ 11 & 11^2 & 11^3 & 11^4 & 11^5 \end{bmatrix}}_{t+1}$$

2) Output $g^{\vec{v}}$.

Evaluate on $Y = \{1,5,6,11\}$ given obfuscation $(g^{v_1}, \dots, g^{v_{t+1}})$

1) Let $\vec{w}$ be random in kernel of

$$M_{t,Y} = \underbrace{\begin{bmatrix} 1 & 1^2 & 1^3 & 1^4 & 1^5 \\ 5 & 5^2 & 5^3 & 5^4 & 5^5 \\ 6 & 6^2 & 6^3 & 6^4 & 6^5 \\ 11 & 11^2 & 11^3 & 11^4 & 11^5 \end{bmatrix}}_{t+1}$$

2) Output 1 iff $g^{\vec{v} \cdot \vec{w}} = g^0$.

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathrm{rowspace}\left(M_{t,X}\right)$

Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathrm{kernel}(M_{t,Y})$

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}(M_{t,X})$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for "Small Superset"

- Use bilinear map $e: G_1 \times G_2 \rightarrow G_T$

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}\left(M_{t,X}\right)$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for "Small Superset"

- Use bilinear map $e: G_1 \times G_2 \to G_T$
- Master Secret Key: random square matrix R

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}(M_{t,X})$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for "Small Superset"

- Use bilinear map $e: G_1 \times G_2 \rightarrow G_T$
- Master Secret Key: random square matrix R
- Public Key: g_2^R

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}\left(M_{t,X}\right)$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for "Small Superset"

- Use bilinear map $e: G_1 \times G_2 \rightarrow G_T$
- Master Secret Key: random square matrix R
- Public Key: g_2^R
- Function decryption key for $f_{t,X}$: $g_1^{\vec{v}\cdot R^{-1}}$

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}\left(M_{t,X}\right)$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for "Small Superset"

- Use bilinear map $e: G_1 \times G_2 \rightarrow G_T$
- Master Secret Key: random square matrix R
- Public Key: g_2^R
- Function decryption key for $f_{t,X}$: $g_1^{\vec{v}\cdot R^{-1}}$
- Encryption of plaintext Y: $g_2^{R\cdot\vec{w}^{\top}}$

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}(M_{t,X})$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for "Small Superset"

- Use bilinear map $e: G_1 \times G_2 \rightarrow G_T$
- Master Secret Key: random square matrix R
- Public Key: g_2^R
- Function decryption key for $f_{t,X}$: $g_1^{\vec{v}\cdot R^{-1}}$
- Encryption of plaintext Y: $g_2^{R\cdot\vec{w}^{\top}}$
- Decryption: Check if $e\left(g_1^{\vec{v}\cdot R^{-1}}, g_2^{R\cdot\vec{w}^{\top}}\right) = 1 \in G_T$

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}\left(M_{t,X}\right)$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for "Small Superset"

- Use bilinear map $e: G_1 \times G_2 \rightarrow G_T$
- **Master Secret Key**: random square matrix R
- **Public Key**: g_2^R
- **Function decryption key for** $f_{t,X}$: $g_1^{\vec{v}\cdot R^{-1}}$
- **Encryption of plaintext** Y: $g_2^{R\cdot\vec{w}^{\top}}$
- **Decryption**: Check if $e\left(g_1^{\vec{v}\cdot R^{-1}}, g_2^{R\cdot\vec{w}^{\top}}\right) = 1 \in G_T$

In the generic group model, random R forces "honest" pairing

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}(M_{t,X})$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for "Small Superset"

- Use bilinear map $e: G_1 \times G_2 \to G_T$
- Master Secret Key: random square matrix R
- Public Key: g_2^R
- Function decryption key for $f_{t,X}$: $g_1^{\vec{v}\cdot R^{-1}}$
- Encryption of plaintext Y: $g_2^{R\cdot\vec{w}^\top}$
- Decryption: Check if $e\left(g_1^{\vec{v}\cdot R^{-1}}, g_2^{R\cdot\vec{w}^\top}\right) = 1 \in G_T$

Simulate with random group elements

Obfuscation: $g^{\vec{v}}$ for $\vec{v} \leftarrow \mathbf{rowspace}(M_{t,X})$
Evaluation: Check if $g^{\vec{v}\cdot\vec{w}} = 1$ for $\vec{w} \leftarrow \mathbf{kernel}(M_{t,Y})$

Function-Private Predicate Encryption for “Small Superset”

- Use bilinear map $e: G_1 \times G_2 \rightarrow G_T$
- **Master Secret Key**: random square matrix R
- **Public Key**: g_2^R
- **Function decryption key for $f_{t,X}$**: $g_1^{\vec{v}\cdot R^{-1}}$
- **Encryption of plaintext Y**: $g_2^{R\cdot\vec{w}^{\top}}$
- **Decryption**: Check if $e\left(g_1^{\vec{v}\cdot R^{-1}}, g_2^{R\cdot\vec{w}^{\top}}\right) = 1 \in G_T$

Simulate with random vectors orthogonal to matching keys

Thank You!

Questions?

Slide Artwork by Eysa Lee