Fully Key-Homomorphic Encryption and its Applications

D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, **Valeria Nikolaenko**,
G. Segev, V. Vaikuntanathan, D. Vinayagamurthy

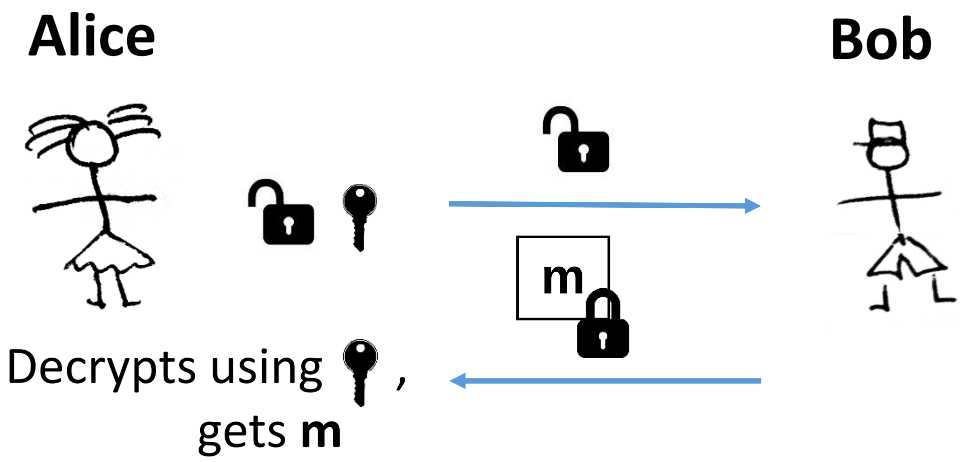
Outline

- Background on PKE and IBE
- Functionality of FKHE
- Applications
- Future & current work

Based on “**Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE and Compact Garbled Circuits**”, EUROCRYPT 2014

Background: Public-Key Encryption (PKE)

[Diffie and Hellman 1976; Rivest, Shamir, Adleman 1977]



Alice | **Bob**

KeyGen → (pk, sk) $\xrightarrow{\text{pk}}$

c = Enc(pk, m)

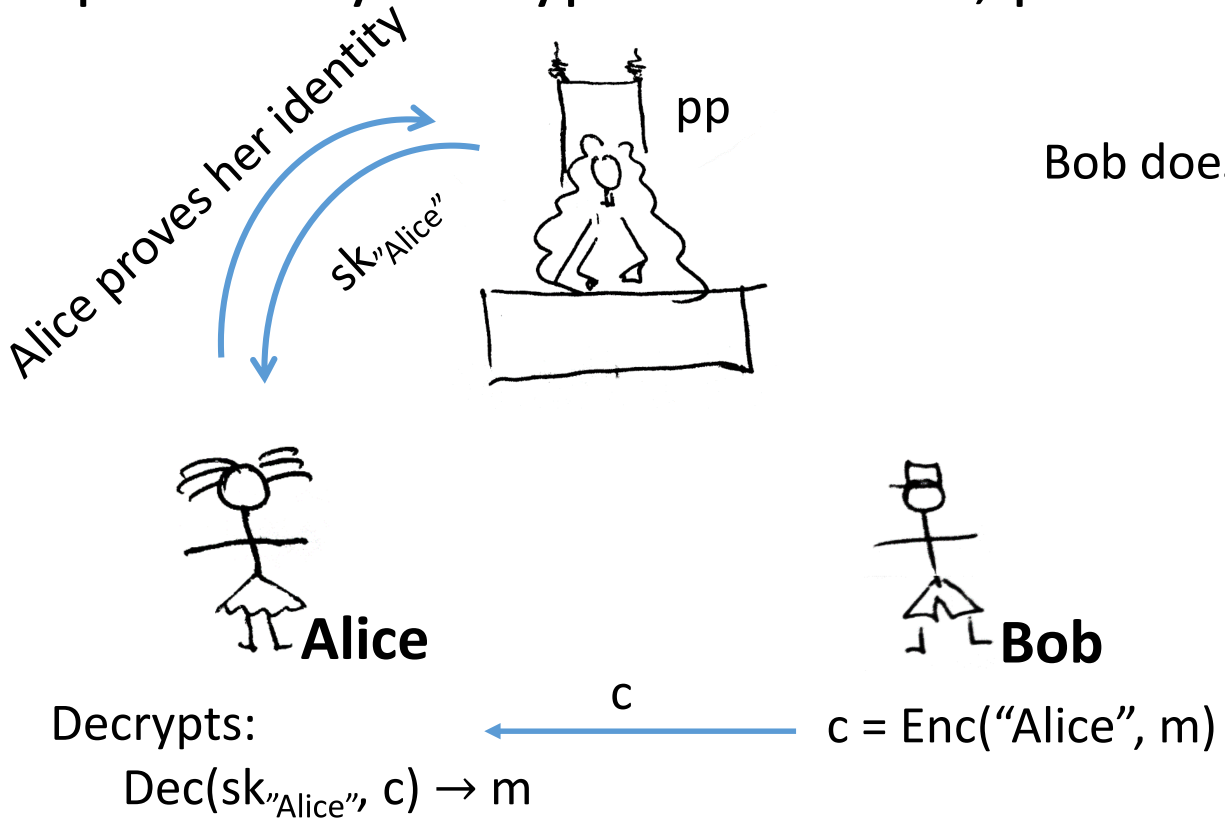
$\xleftarrow{\text{c}}$

Decrypts:
Dec(sk, c) → m

Background: Identity Based Encryption (IBE)

Proposed by Shamir 1984, constructed by Boneh, Franklin and Cocks in 2001

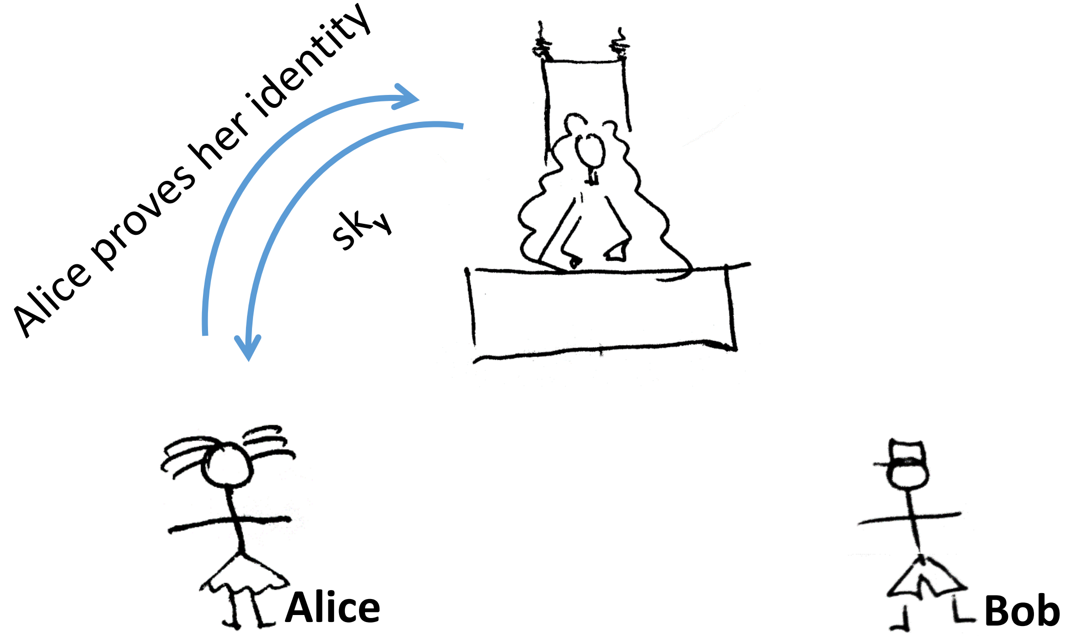
- IBE is a public key encryption scheme; public keys are identities



Bob does not need to know Alice’s public key

Functionality of FKHE

IBE: Identities are **x,y** $\in \mathbb{Z}^k_q$ (attributes)



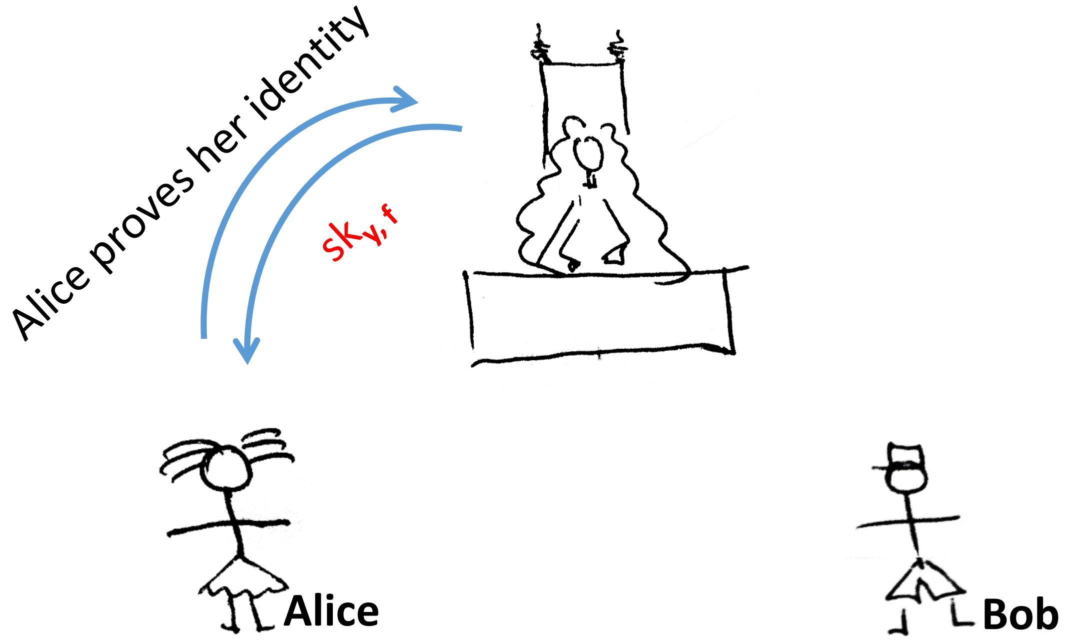
c_x

Decrypts:

$Dec(sk_{\mathbf{y}}, c_x) \rightarrow m$ iff $\mathbf{x} == \mathbf{y}$

$c_x = Enc(\mathbf{x}, m)$

FKHE: $\mathbf{x} \in \mathbb{Z}^k_q$, $f \in \{\mathbb{Z}_q^k \rightarrow \mathbb{Z}_q\}$, $y \in \mathbb{Z}_q$



c_x

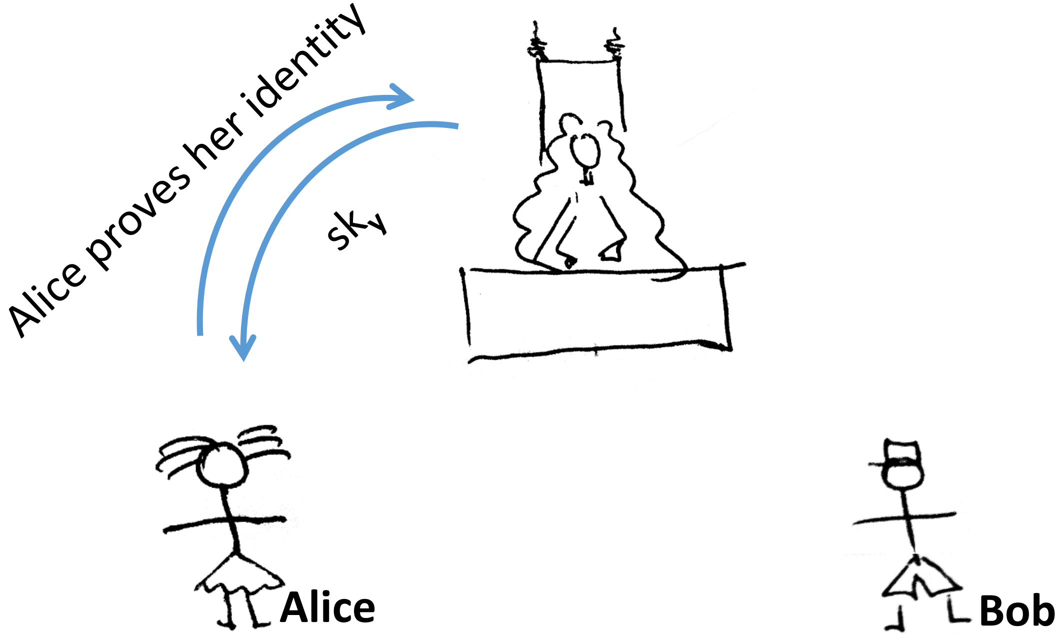
Decrypts:

$Dec(sk_{y,f}, c_x) \rightarrow m$ iff $f(x) == y$

$c_x = Enc(\mathbf{x}, m)$

Functionality of FKHE

IBE: Identities are $\mathbf{x},\mathbf{y} \in \mathbb{Z}^k_q$ (attributes)

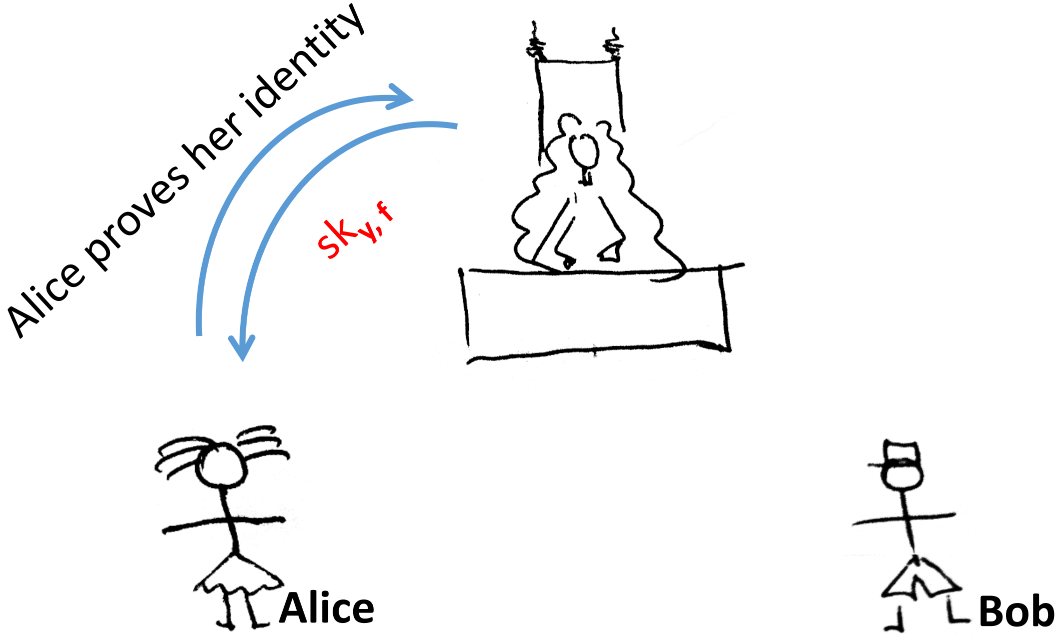


Decrypts:

c_x ← $c_x = \text{Enc}(\mathbf{x}, m)$

$\text{Dec}(sk_y, c_x) \rightarrow m$ iff $\mathbf{x} == \mathbf{y}$

FKHE: $\mathbf{x} \in \mathbb{Z}^k_q$, $f \in \{\mathbb{Z}_q^k \rightarrow \mathbb{Z}_q\}$, $y \in \mathbb{Z}_q$



Evaluates:

c_x ← $c_x = \text{Enc}(\mathbf{x}, m)$

$\text{Enc}(x, m) \rightarrow \text{Enc}(\langle f(x), f\rangle, m)$

$c_x \rightarrow c_{f(x), f}$

Key Homomorphism

Decrypts:

$\text{Dec}(sk_{\mathbf{y,f}}, c_{f(x),f}) \rightarrow m$ iff $f(x) == y$

Functionality of FKHE II

- **Secure under LWE**
- **Unbounded number of collusions**

- **Setup**(1^λ) → pp, msk
- **KeyGen**(msk, (y, f)) → $sk_{y,f}$

  Secret key for pk = (y, f)

  $\in \mathbb{Z}_q$ $\quad \in \{\mathbb{Z}_q^k \to \mathbb{Z}_q\}$

- **Enc**(pp, x, m) → c_x

  Encryption under pk = x

  $\in \mathbb{Z}^k{}_q$

- **Eval**(pp, f, c_x) → $c_{f(x), f}$

  Encryption under pk = (f(x), f)

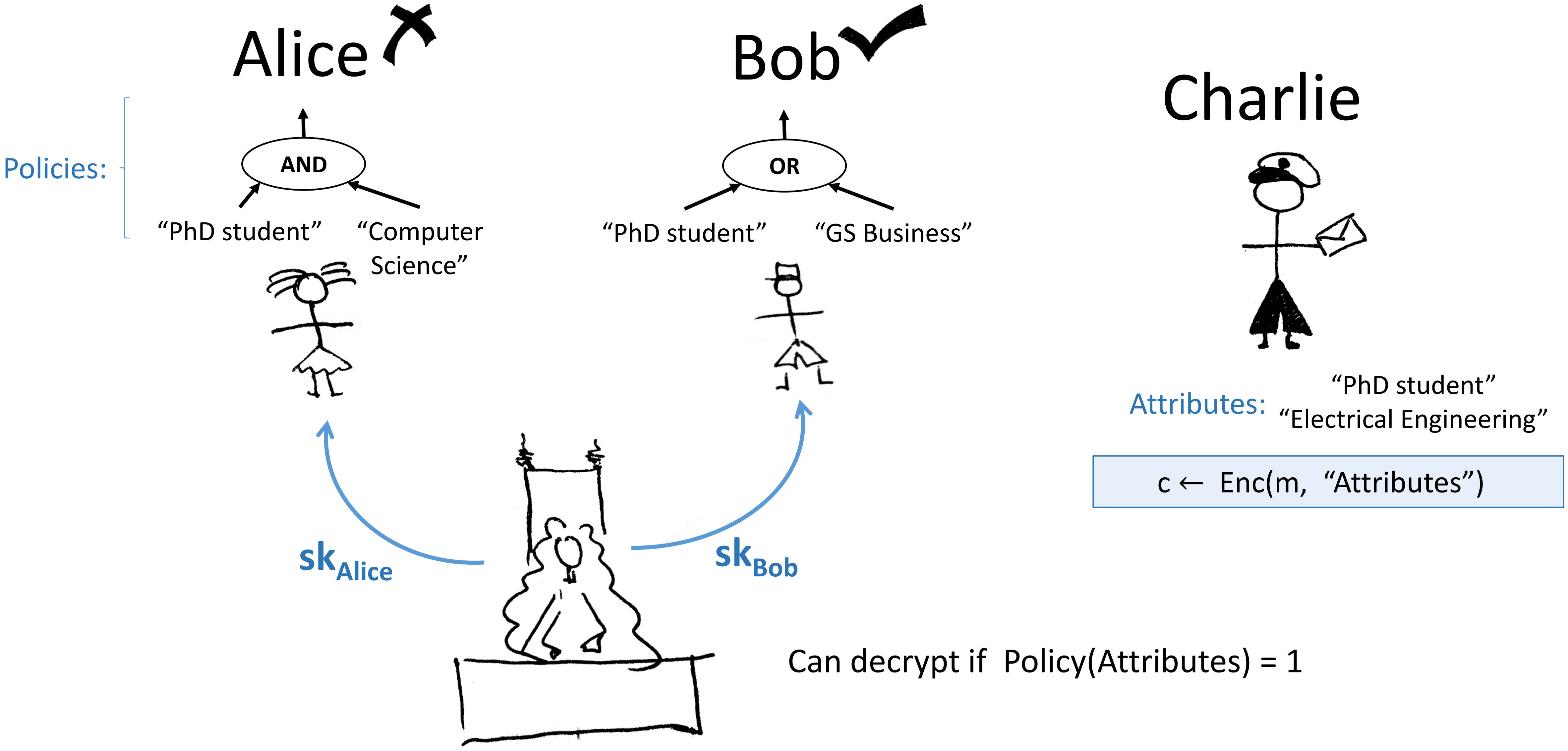
  $\in \{\mathbb{Z}_q^k \to \mathbb{Z}_q\}$

- **Dec**($c_{f(x), f}$, $sk_{y, f}$) → m iff f(x) = y

Applications:

- Attribute Based Encryption: short secret keys, arithmetic circuits
- Compressed Garbled Circuits

Attribute Based Encryption (ABE) [SW05]



Attribute Based Encryption

	Polices	Attributes	Security	Key size
SW05	Single threshold gates	In ciphertext	BDH	
GPSW06, HW13	Monotone formulas	In key	BDH	O(size)
BSW07	Monotone formulas	In ciphertext	Non-standard assumption	O(size)
GJPS08	Bounded size threshold gts	In ciphertext	DBDH	
LOSTW10	Monotone formulas	In ciphertext	Non-standard assumption	O(size)
OT10	Span programs	In key	DLIN	
ABVVW12	Single threshold	In cipher	LWE	
Wat12	DFA	In key	l-Expanded BDHE	
OSW12	Any formula	In key	DBDH	O(size)
SW12, GGH12, GGHSW13	Boolean circuits	In key, in cipher	Multi-linear maps	O(size)
Boy13	Boolean formulas	In key	LWE	O(size)
GVW13	Boolean circuits	In key	LWE	O(size)
This	**Arithmetic circuits**	**In key**	**LWE**	**O(depth)**

ABE from FKHE

Alice

Policy: f

Bob

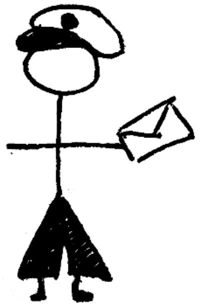
Policy: g

Charlie

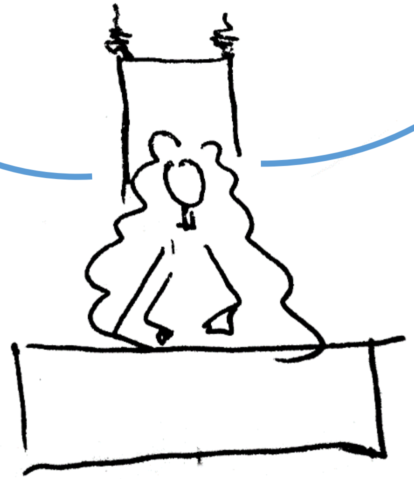
Attribute: x

Message: m

Ciphertext: $c_x \leftarrow \mathrm{Enc}(pp, x, m)$



$\mathbf{sk_{Alice} = sk_{1,f}}$



$\mathbf{sk_{Bob} = sk_{1,g}}$

$\mathrm{KeyGen}(msk, 1, f) \rightarrow sk_{1,f}$

$\mathrm{KeyGen}(msk, 1, g) \rightarrow sk_{1,g}$

ABE from FKHE (decryption)

Charlie

Alice (has $sk_{1,f}$)

$c_x \longleftarrow \mathrm{Enc}(x, m)$

$\mathrm{Eval}(\mathbf{f}, \mathbf{c_x})$

$c_{f(x),f} = \mathrm{Enc}(\mathbf{(f(x), f)}, m)$

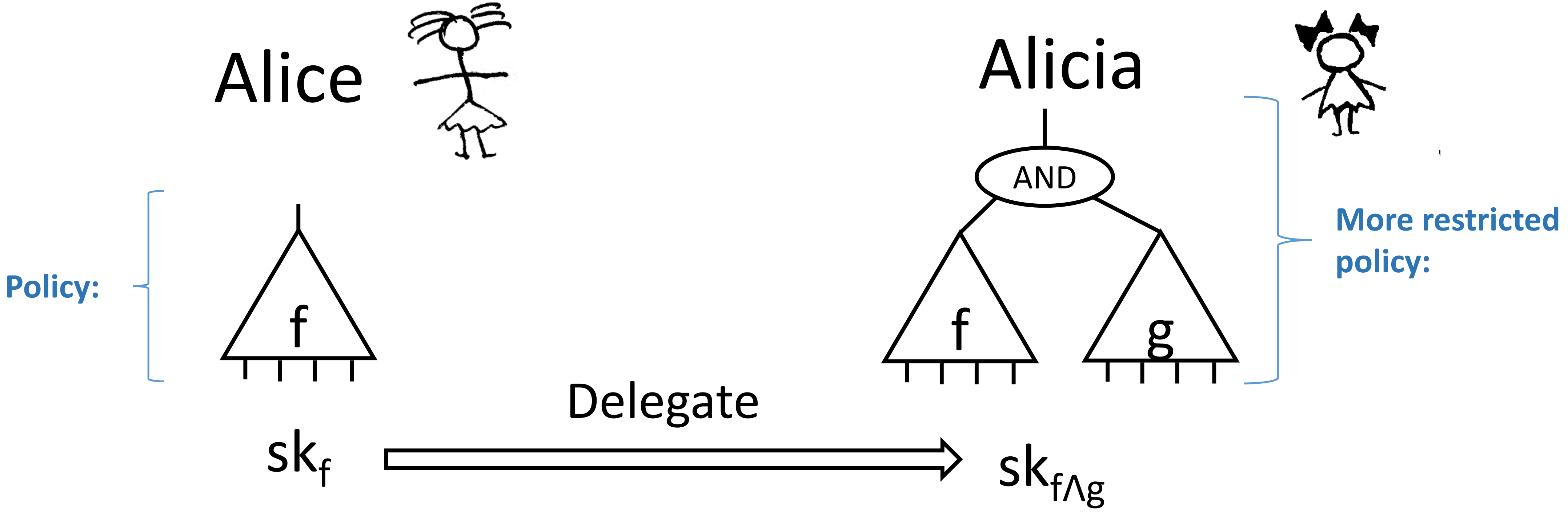
If $f(x) = 1$ can decrypt with $sk_{1,f}$

<u>Thm</u>: **FKHE** is secure $\Rightarrow$ **ABE** is secure

Our new ABE (key policy)

- Arithmetic circuits, not just boolean
- Key size depends on <u>depth</u>, not on size
- Arbitrary fan-in gates
- Delegatable

Delegation



Our scheme supports delegation:

Alice can create a custom restricted secret key **herself.**

Garbled Circuits

Want to compute f(x, y), not revealing x or y

[Yao'86]:

Alice (x)

Bob (y)

garbled circuit: $O(|f|)$

For new y', repeat

OT: $O(|x| + |y|)$

f(x, y)

[GKPVZ'13]: reusable garbled circuit

Using our ABE:
becomes short O(depth)

Alice (x)

Bob (y)

garbled circuit: $O(|f|)$

$O(|x| + |y|)$

For new y', repeat

f(x, y)

Garbled Circuits

[GKPVZ'13] [This]

ABE **with short keys**

FHE

Garbled circuits

$\Longrightarrow$ Reusable **compressed** garbled circuits

O(depth)

Future & current work

FKHE hides neither x nor f

Functional Encryption | **Hiding x**

Hiding f | Program Obfuscation

Alice

sk_f

$c_x = Enc(x)$

Bob

$Dec(c_x, sk_f) \to f(x)$

In FHE Bob gets $c_{f(x)} = Enc(\ f(x)\ )$

Alice

$\Theta(f) = sk_f$

Bob

- Can compute f(x) for any x
- Learns “nothing” about f

Thank you

valerini@stanford.edu