TCC 2022 Nov 8, 2022

IBE with Incompressible Master Secret and Small Identity Secrets

Sruthi Sekar

Berkeley | EECS

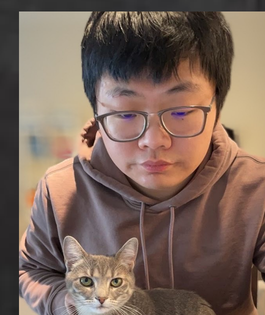
Nico Döttling

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Sanjam Garg

Berkeley | EECS

NTT Research



Mingyuan Wang

Berkeley | EECS

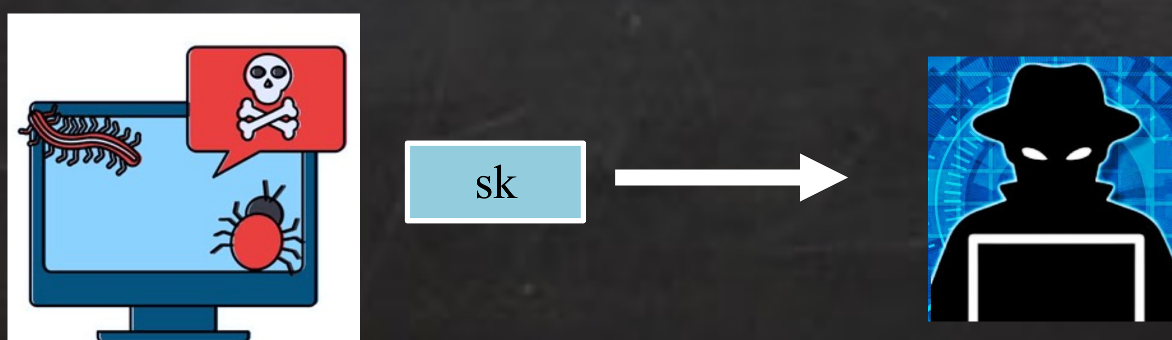
Big-key Cryptography

Motivation: Exfiltration Attacks

Big-key Cryptography

Motivation: Exfiltration Attacks

- Computers with valuable cryptographic keys prone to exfiltration attacks–giving adversary unrestricted access!



Big-key Cryptography

Motivation: Exfiltration Attacks

- Computers with valuable cryptographic keys prone to exfiltration attacks– giving adversary unrestricted access!

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute

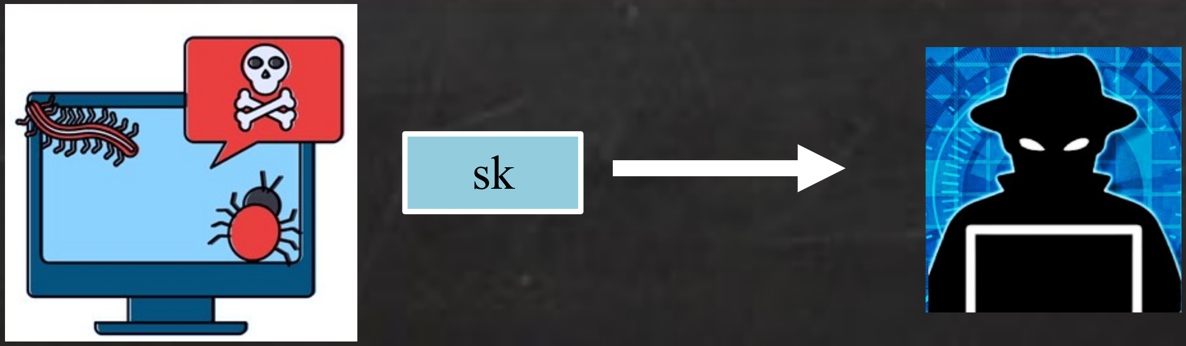
Article | Talk | Read | Edit | View histo

Advanced persistent threat

From Wikipedia, the free encyclopedia

An **advanced persistent threat** (**APT**) is a stealthy threat actor, typically a nation state or state-sponsored group, which gains unauthorized access to a computer network and remains undetected for an extended period.[1][2] In recent times, the term may also refer to non-state-sponsored groups conducting large-scale targeted intrusions for specific goals.[3]

Such threat actors' motivations are typically political or economic.[4] Every major business sector has recorded instances of cyberattacks by advanced actors with specific goals, whether to steal, spy, or disrupt. These targeted sectors include government, defense, financial services, legal services, industrial, telecoms, consumer goods and many more.[5][6][7] Some groups utilize traditional espionage vectors, including social engineering, human intelligence and infiltration to gain access to a physical location to enable network attacks. The purpose of these attacks is to install custom malware (malicious software).[8]



Big-key Cryptography

Motivation: Exfiltration Attacks

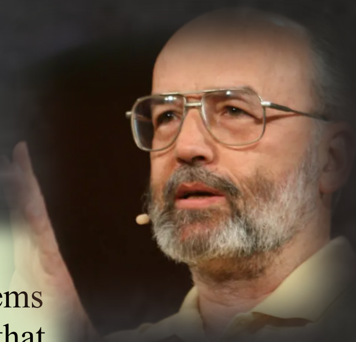
- Computers with valuable cryptographic keys prone to exfiltration attacks– giving adversary unrestricted access!
- Solution: Make the secret key BIG (say ~ 1 TB)!

sk

Big-key Cryptography

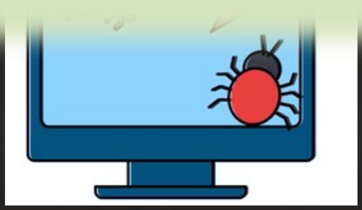
Motivation: Exfiltration Attacks

- Computers with valuable cryptographic keys prone to exfiltration attacks–giving adversary unrestricted access!
- Solution: Make the secret key BIG (say ~ 1 TB)!



Adi Shamir
@RSA 2013

“We have to think in a totally different way about how we are going to protect computer systems assuming there are APTs inside already which cannot be detected. Is everything lost? I claim that not: there are many things that you can do, because the APT is basically going to have a very, very narrow pipeline to the outside world. . . . I would like, for example, all the small data to become big data, just in terms of size. I want that the secret of the Coco-Cola company to be kept not in a tiny file of one kilobyte, which can be exfiltrated easily by an APT · · · . I want that file to be a terabyte, which cannot be [easily] exfiltrated.”



sk

Big-key Cryptography

Bounded Retrieval Model

- Computers with valuable cryptographic keys prone to exfiltration attacks– giving adversary unrestricted access!

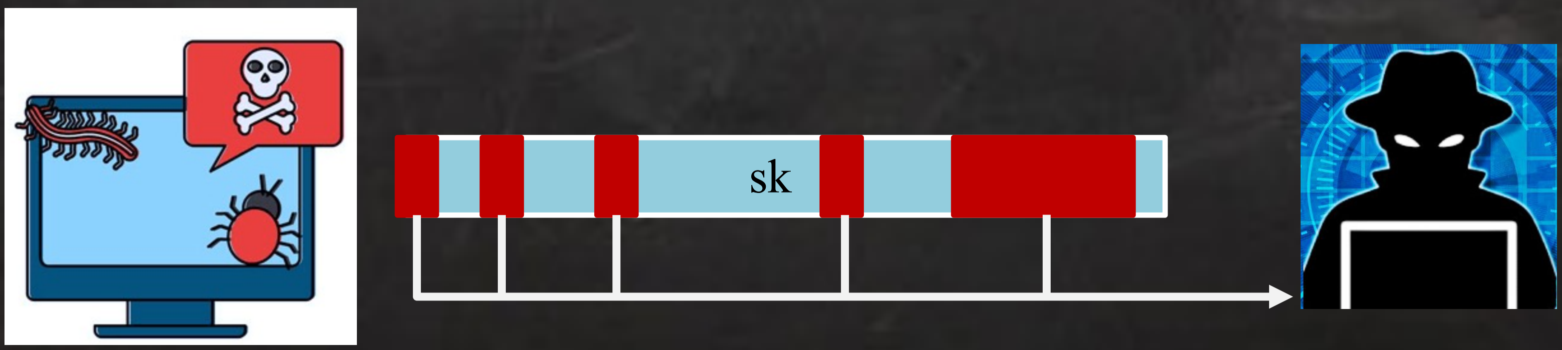
- Solution: Make the secret key BIG (say ~ 1 TB)!

- Big-key Cryptosystems [Dzi06, DLW06, CDD+07, ADW09, ADN+10, BKR16, MW20] (in bounded-retrieval model (BRM)):

Big-key Cryptography

Bounded Retrieval Model

- Computers with valuable cryptographic keys prone to exfiltration attacks– giving adversary unrestricted access!
- Solution: Make the secret key BIG (say ~ 1 TB)!
- Big-key Cryptosystems [Dzi06, DLW06, CDD+07, ADW09, ADN+10, BKR16, MW20] (in bounded-retrieval model (BRM)):
  - ➢ Leakage: Adversary gets arbitrary bounded bits of the secret key (say even 99% of the key is leaked).



Big-key Cryptography

Bounded Retrieval Model

- Computers with valuable cryptographic keys prone to exfiltration attacks– giving adversary unrestricted access!
- Solution: Make the secret key BIG (say ~ 1 TB)!
- Big-key Cryptosystems [Dzi06, DLW06, CDD+07, ADW09, ADN+10, BKR16, MW20] (in bounded-retrieval model (BRM)):
    - Leakage: Adversary gets arbitrary bounded bits of the secret key (say even 99% of the key is leaked)
    - Efficiency: Cryptosystem shouldn't read entire big-key to preserve efficiency (through locality).

Big-key Cryptography

Bounded Retrieval Model

- Computers with valuable cryptographic keys prone to exfiltration attacks– giving adversary unrestricted access!

- Solution: Make the secret key BIG (say ~ 1 TB)!

- Big-key Cryptosystems [Dzi06, DLW06, CDD+07, ADW09, ADN+10, BKR16, MW20] (in bounded-retrieval model (BRM)):

    - Leakage: Adversary gets arbitrary bounded bits of the secret key (say even 99% of the key is leaked)
    - Efficiency: Cryptosystem shouldn't read entire big-key to preserve efficiency (through locality).

Big-key primitives: symmetric key encryption [BKR16], public-key encryption [ADN+10, MW20], authenticated key agreement [Dzi06, CDD+07, ADW09].

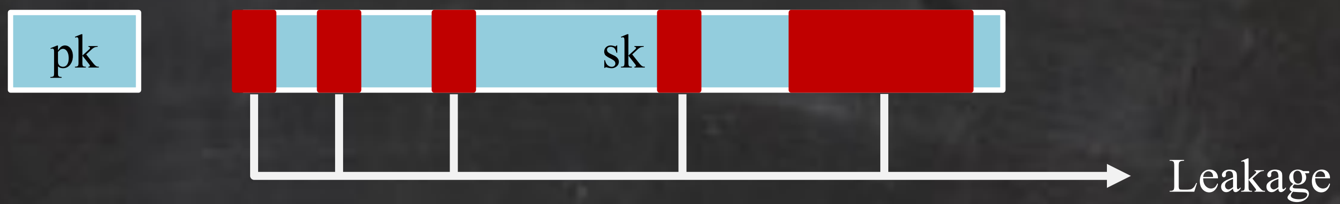
Big-key (Public-key) Encryption

The Model

Big-key (Public-key) Encryption

The Model

- Public-key setup: small public key, large secret key (prone to exfiltration).



Big-key (Public-key) Encryption

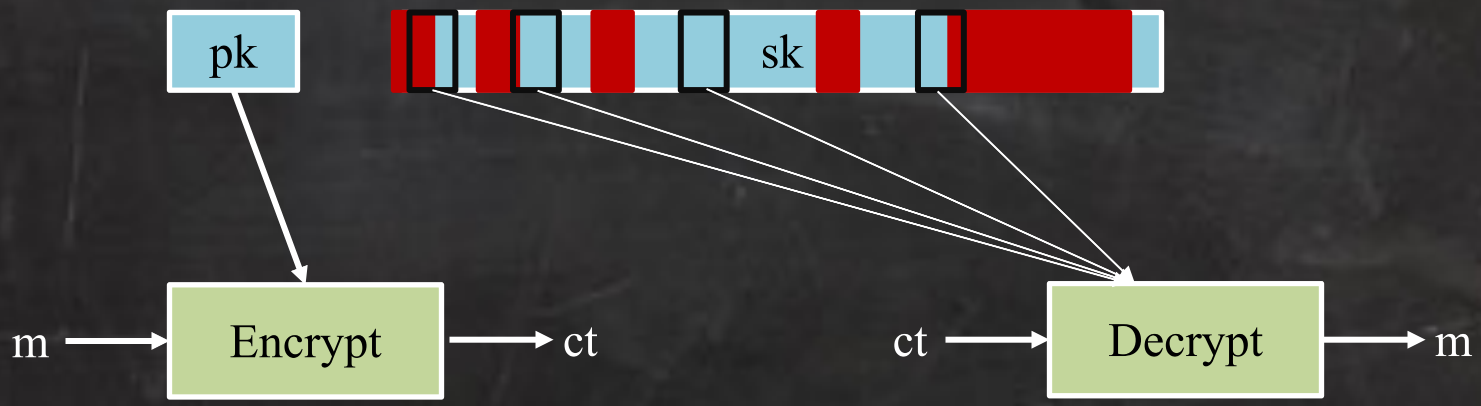
The Model

- Public-key setup: small public key, large secret key (prone to exfiltration).
- Encryption/Decryption: running times don't grow with size of big secret key

Big-key (Public-key) Encryption

The Model

- Public-key setup: small public key, large secret key (prone to exfiltration).
- Encryption/Decryption: running times don't grow with size of big secret key –decryption makes local ciphertext-dependent access to the secret key.

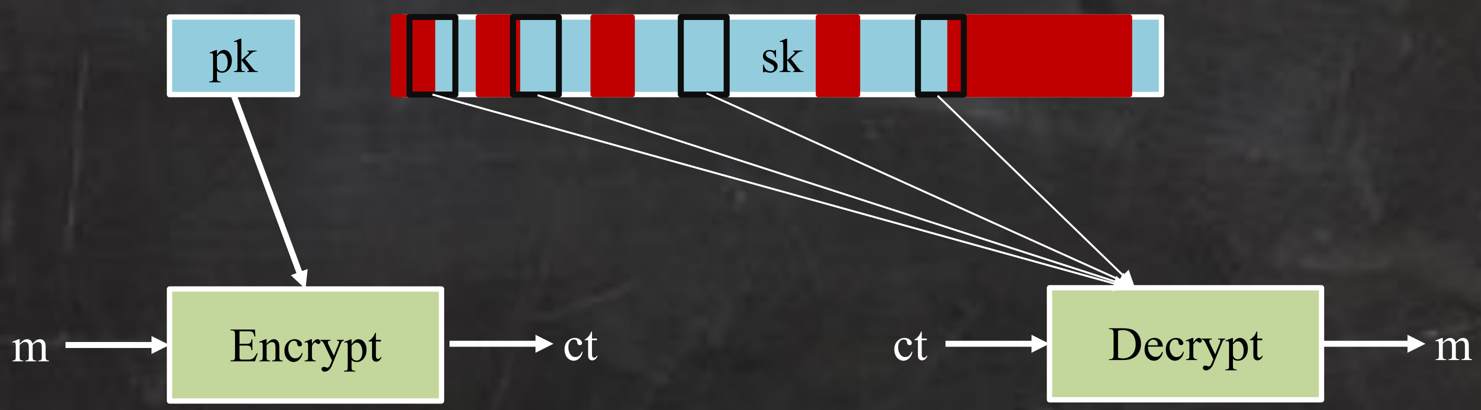


Big-key (Public-key) Encryption

The Model

- Public-key setup: small public key, large secret key (prone to exfiltration).
- Encryption/Decryption: running times don't grow with size of big secret key –decryption makes local ciphertext-dependent access to the secret key.

Security: semantic security of fresh ciphertexts generated after arbitrary leakage on sk is given to the adversary.



Big-key (Public-key) Encryption

Caveats of The Model

Big-key (Public-key) Encryption

Caveats of The Model

User must carry entire large secret key on all its devices.
(since parts of sk needed to decrypt are unknown a priori)

Big-key (Public-key) Encryption

Caveats of The Model

User must carry entire large secret key on all its devices.

1. Leads to wastage of limited storage space on small mobile devices.

Big-key (Public-key) Encryption

Caveats of The Model

User must carry entire large secret key on all its devices.

1. Leads to wastage of limited storage space on small mobile devices.

2. Replication of the large secret makes use more susceptible to leakage (e.g., the loss of a mobile device will leak whole of sk!)

Our Solution: Big-key IBE

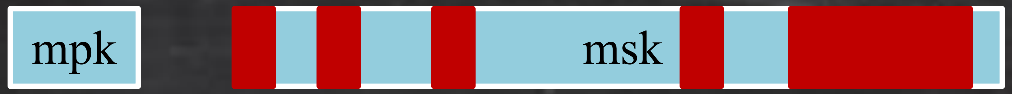
Our IBE-based Model

Our Solution: Big-key IBE

Our IBE-based Model

Use the advantages of identity-based encryption-

- Setup: generates master public and secret keys (mpk,msk).
  (Big key setup: msk is now a big-key and prone to exfiltration).



Our Solution: Big-key IBE

Our IBE-based Model

Use the advantages of identity-based encryption-

- Setup: generates master public and secret keys (mpk,msk).
  (Big key setup: msk is now a big-key and prone to exfiltration).

- Encryption relies only on short mpk, public identity id, and message m.

Our Solution: Big-key IBE

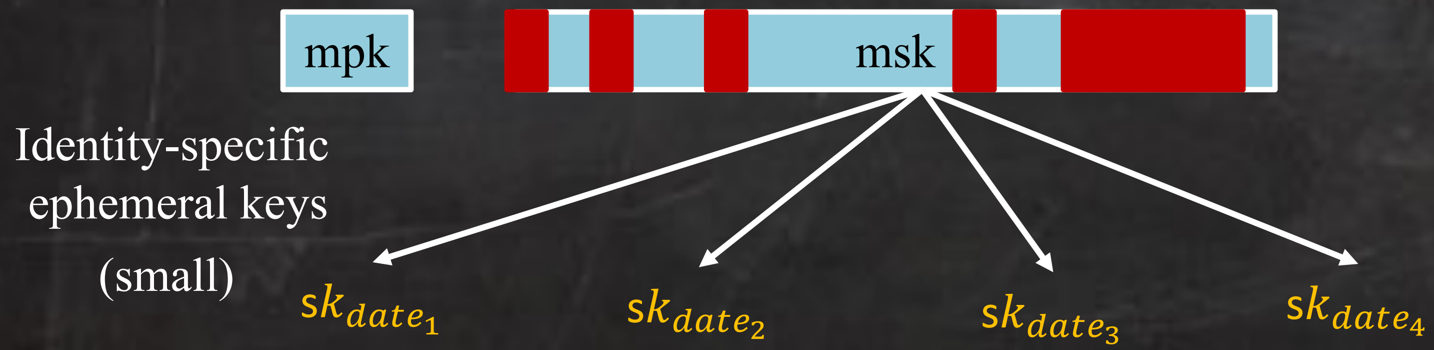
Our IBE-based Model

Use the advantages of identity-based encryption-

- Setup: generates master public and secret keys (mpk,msk).
  (Big key setup: msk is now a big-key and prone to exfiltration).

- Encryption relies only on short mpk, public identity id, and message m.

- Decryption uses short identity-specific ephemeral keys sk_{id}.



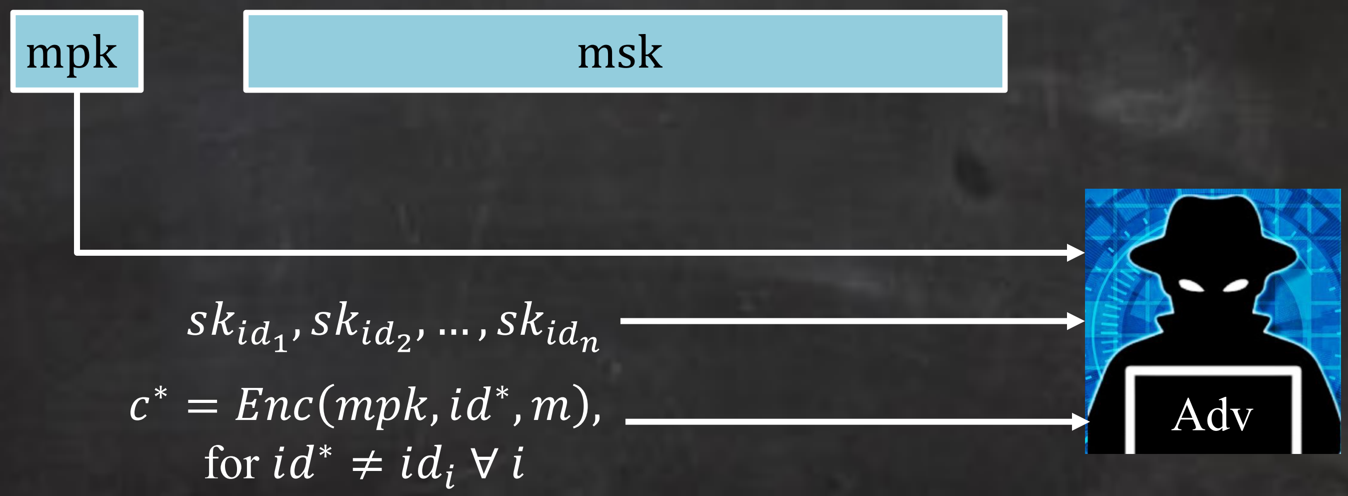
Big-key IBE

Challenges in Defining Security

Big-key IBE

Challenges in Defining Security

- Standard IBE security: Adv gets polynomial number of sk_{id}'s, challenge ciphertext $c^* = Enc(\text{mpk}, id^*, m)$ hides m.



Big-key IBE

Challenges in Defining Security

- Standard IBE security: Adv gets polynomial number of sk_{id}'s, challenge ciphertext $c^* = Enc(\text{mpk}, id^*, m)$ hides m.

mpk　　msk

$sk_{id_1}, sk_{id_2}, \ldots, sk_{id_n}$

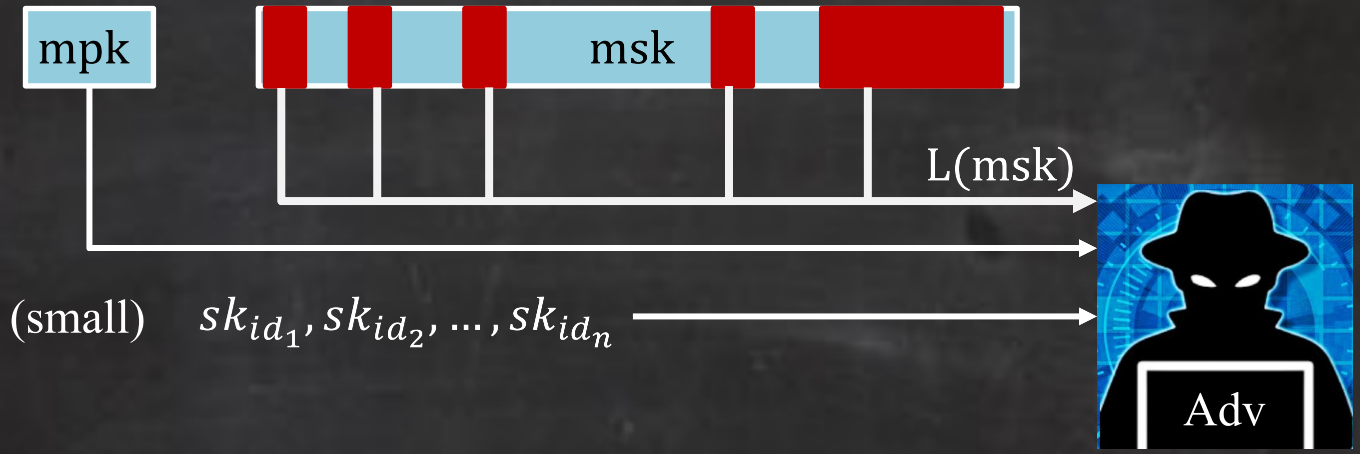
$c^* = Enc(mpk, id^*, m)$,
for $id^* \neq id_i \; \forall \; i$

Selective security: id^* given by Adv before seeing mpk.
Full security: id^* given by Adv after seeing mpk.

Big-key IBE

Challenges in Defining Security

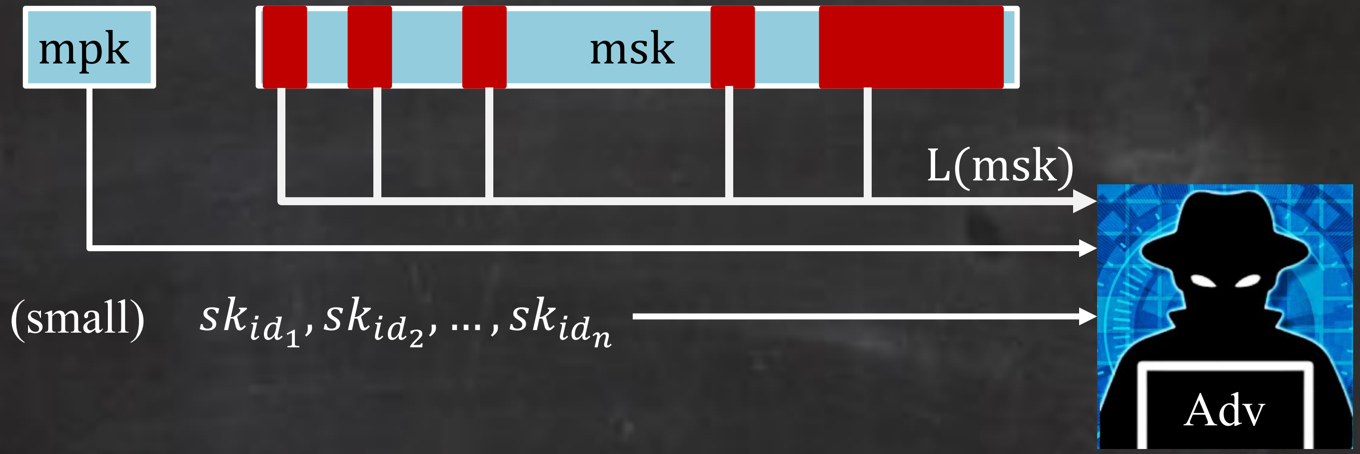
- Big-key IBE security: Adv gets L(msk) in addition to polynomial number of sk_{id}'s.



Big-key IBE

Challenges in Defining Security

- Big-key IBE security: Adv gets L(msk) in addition to polynomial number of sk_{id}'s.

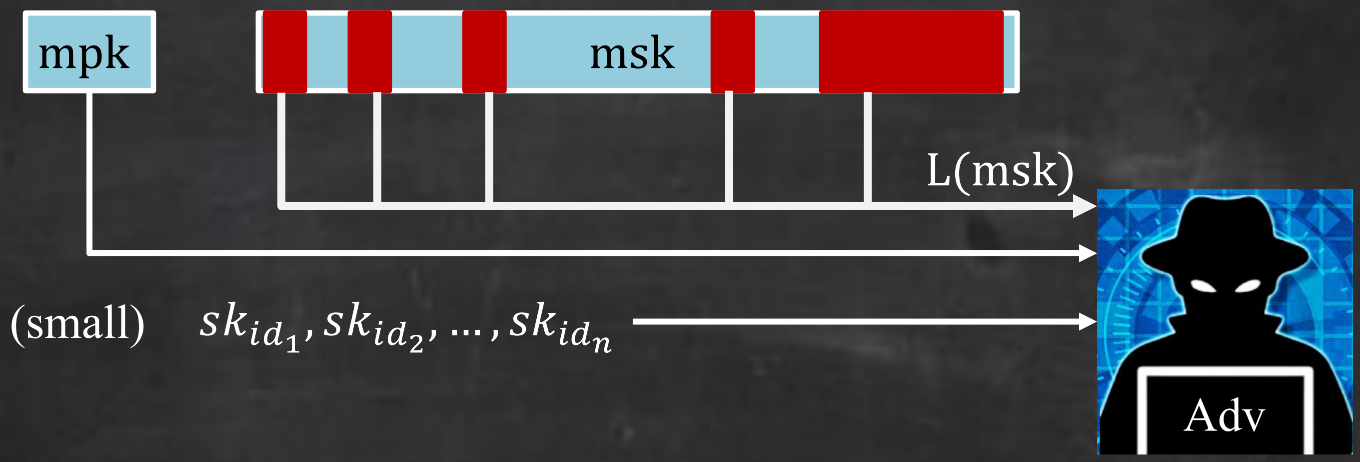


Key Challenge in defining security: Adv can get the challenge sk_{id^*} directly through L(msk) (since the output length of L is large)—breaks security.

Big-key IBE

Challenges in Defining Security

- Big-key IBE security: Adv gets L(msk) in addition to polynomial number of sk_{id}'s.



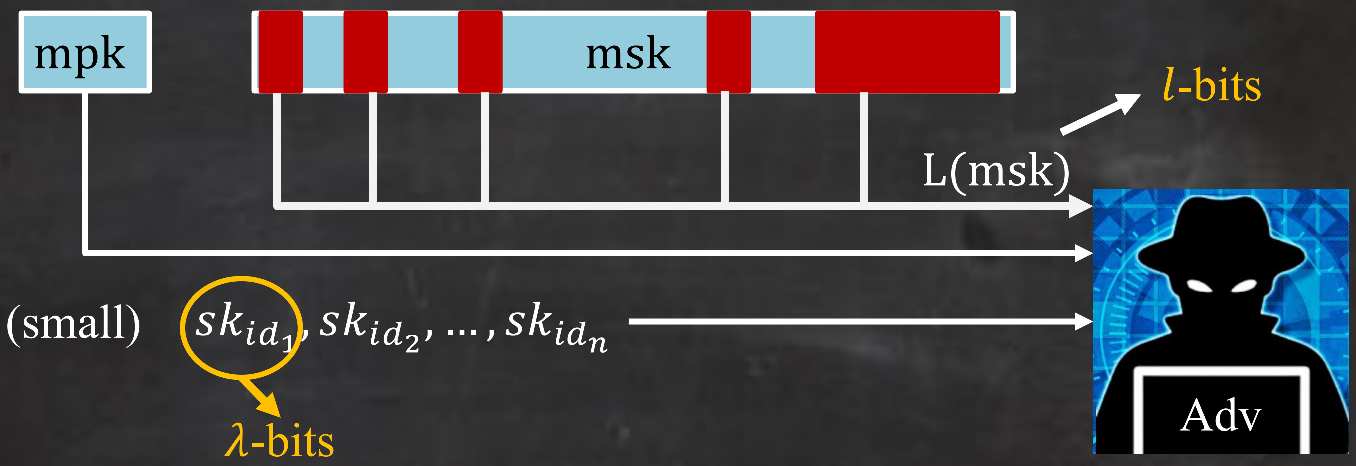
Key Challenge in defining security: Adv can get the challenge sk_{id^*} directly through L(msk) (since the output length of L is large)—breaks security.

Prior Leakage-resilient IBEs [ADN+10, CDRW10,LRW11,HLWW13,CZLC16,NY19] —had large sk_{id}'s and msk is either large or allows no leakage.

Big-key IBE

Towards Defining Security

- Big-key IBE security: Adv gets L(msk) in addition to polynomial number of sk_{id}'s.

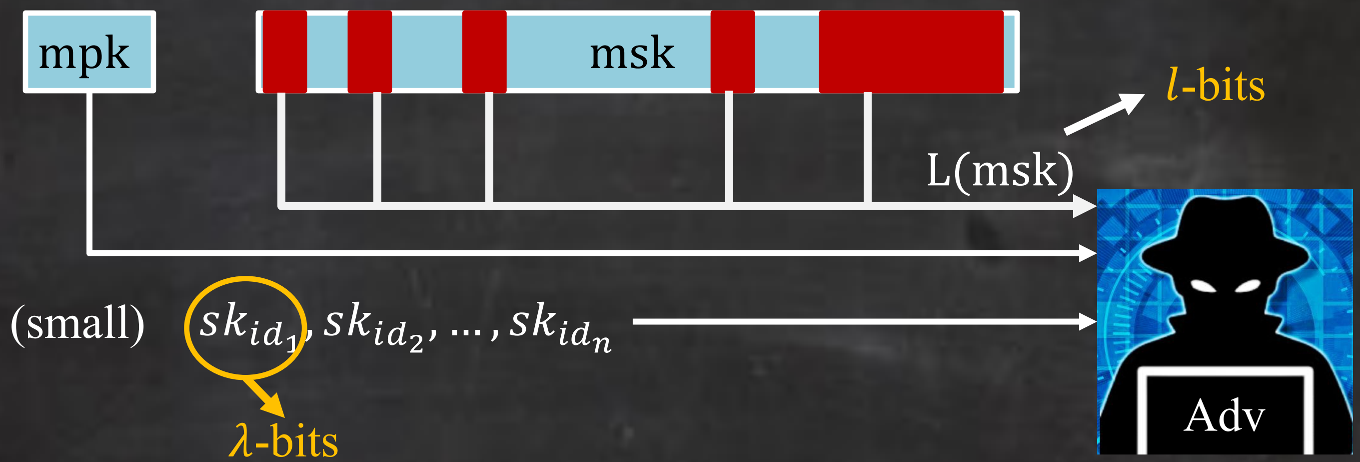


- Adv through the l-bit leakage allowed by L can get upto $\frac{l}{\lambda} = \Theta(l)$ sk_{id}'s.

Big-key IBE

Towards Defining Security

- Big-key IBE security: Adv gets L(msk) in addition to polynomial number of sk_{id}'s.

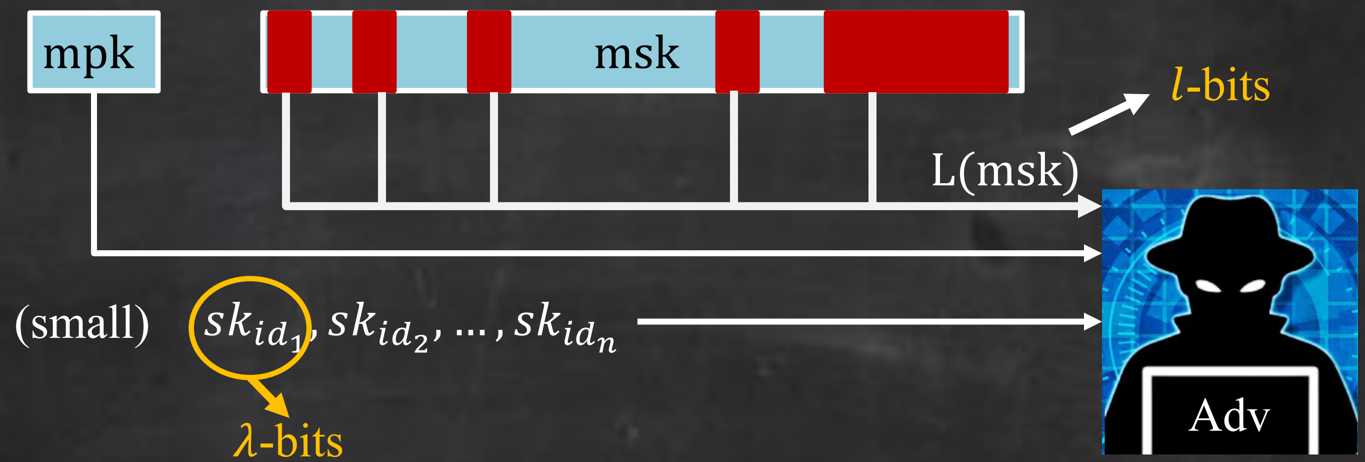


- Adv through the l-bit leakage allowed by L can get upto $\frac{l}{\lambda} = \Theta(l)\ sk_{id}$'s.
- Intuition for our definition: Adv should not get more information than what he would get via the above *trivial exfiltration attack.*

Big-key IBE

Towards Defining Security

- Big-key IBE security: Adv gets L(msk) in addition to polynomial number of sk_{id}'s.



- Adv through the l-bit leakage allowed by L can get upto $\frac{l}{\lambda} = \Theta(l)$ sk_{id}'s.
- Intuition for our definition: Adv should not get more information than what he would get via the above *trivial exfiltration attack.*
  Particularly, given l-bit leakage, we want Adv to not break security for $\geq l + 1$ identities.

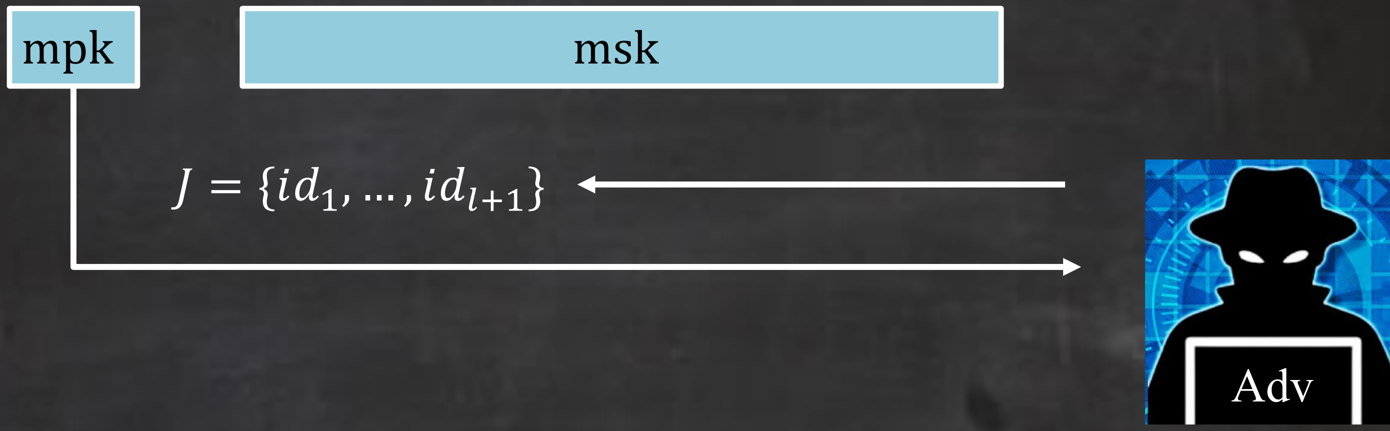
Big-key IBE

Our Security Model

$J = \{id_1, \ldots, id_{l+1}\}$

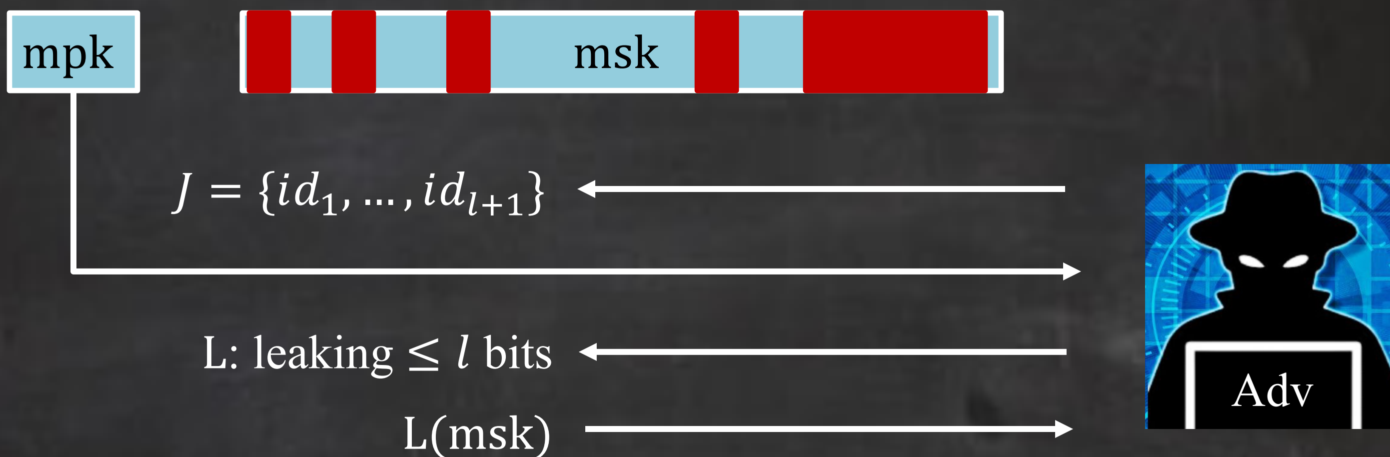
Big-key IBE

Our Security Model



Big-key IBE

Our Security Model



Big-key IBE

Our Security Model

mpk

msk

$J = \{id_1, \dots, id_{l+1}\}$

L: leaking $\leq l$ bits

L(msk)

Exp^{id} (for $id \in J$):

Big-key IBE

Our Security Model

mpk

msk

$J = \{id_1, \ldots, id_{l+1}\}$

L: leaking $\leq l$ bits

L(msk)

Exp^{id} (for $id \in J$):

L(msk), mpk, ct, u^b

Big-key IBE

Our Security Model

mpk

msk

$J = \{id_1, \ldots, id_{l+1}\}$

L: leaking $\leq l$ bits

L(msk)

Exp^{id} (for $id \in J$):

$\text{L(msk), mpk}, ct, u^b$

$ct, u^0 \leftarrow Encap(id)$

$u^1 \in_R$

Big-key IBE

Our Security Model

mpk

msk

$J = \{id_1, \ldots, id_{l+1}\}$

L: leaking $\leq l$ bits

L(msk)

Exp^{id} (for $id \in J$):

L(msk), mpk, ct, u^b

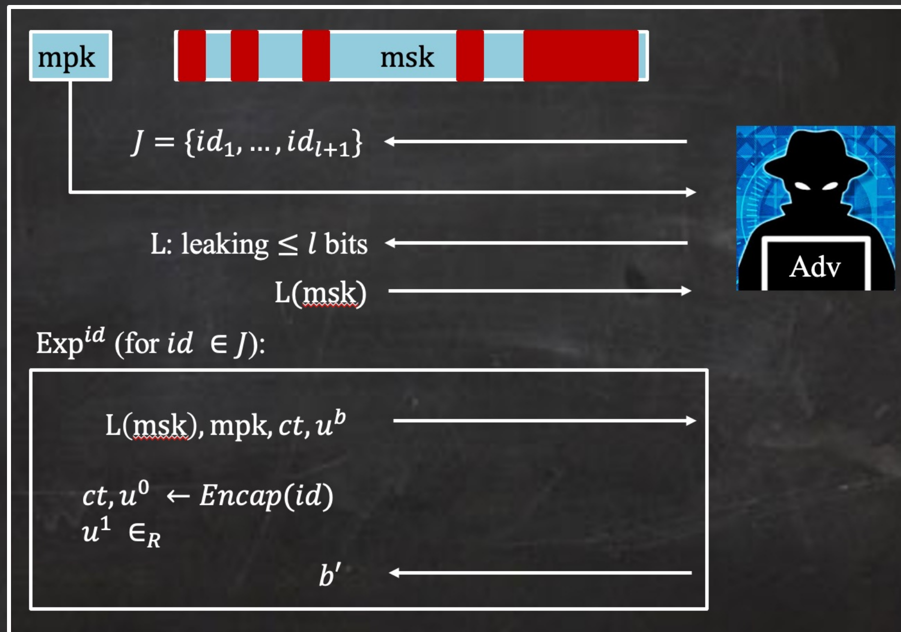
$ct, u^0 \leftarrow Encap(id)$

$u^1 \in_R$

b'

Big-key IBE

Our Security Model



Selective Security:

$$\Pr[\forall\, id \in J, |\Pr[b' = b] - 1/2| \geq \varepsilon]$$ is negligible.

Big-key IBE

Construction Overview

Big-key IBE

Construction Overview

Big-key
Pseudo-entropy
Function

Big-key IBE

Construction Overview

Big-key Pseudo-entropy Function

+

Leakage-resilient Encryption [HLWW13]
Reusable 2-round MPC [BL20]

Big-key IBE with large mpk

Big-key IBE

Construction Overview

Big-key Pseudo-entropy Function

+

Leakage-resilient Encryption [HLWW13]
Reusable 2-round MPC [BL20]

Big-key IBE with large mpk

+

Laconic OT [CDG+17]

Big-key IBE

*Security relies on hardness of standard assumptions on groups with bilinear pairing

Open Problems

- How to make the construction black-box in the underlying primitives?
- How to make the big-keys useful, a.k.a. catalytic?
  (as in the work of [MW20])

Open Problems

- How to make the construction black-box in the underlying primitives?
- How to make the big-keys useful, a.k.a. catalytic?
  (as in the work of [MW20])

THANK YOU

eprint/2022/649