Backward-Leak Uni-Directional Updatable Encryption from (Homomorphic) Public Key Encryption

Yao Jiang Galteland and **Jiaxin Pan**

NTNU Trondheim, Norway

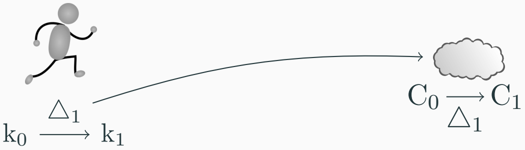
NTNU
Norwegian University of
Science and Technology

Our Main Contributions

- Backward-leak uni-directional key update setting $\Leftrightarrow$ No-directional one
- Two Generic Constructions of UE
  - From homomorphic PKE (concurrent to Miao, Patranabis, Watson, PKC 2023)
  - From bootstrappable PKE

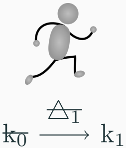
- **Unidirectional Updatable Encryption and Proxy Re-encryption from DDH**
  Miao, Patranabis, Watson; PKC 2023 (ePrint 2022/311)

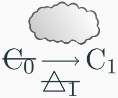
Recall: Updatable Encryption (UE)



- **Key Homomorphic PRFs and their Applications**
  Boneh, Lewi, Montgomery, Raghunathan; CRYPTO '13 (ePrint 2015/220)

Recall: Updatable Encryption (UE)





- Client only needs to store one key
- Security (informal): (freshly encrypted or updated) ciphertexts or tokens should leak nothing about the plaintext

- **Key Homomorphic PRFs and their Applications**
  Boneh, Lewi, Montgomery, Raghunathan; CRYPTO '13 (ePrint 2015/220)

Epoch-based Model

$\texttt{time} \longrightarrow$

0	1	2	3	4	5	6	7	...	n
	$\triangle_1$	$\triangle_2$	$\triangle_3$	$\triangle_4$	$\triangle_5$	$\triangle_6$	$\triangle_7$	...	
k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	...	k_n
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	...	C_n

Bi-directional Updates

0	1	2	3	...	i-1	i	...	n
	$\triangle_1$	$\triangle_2$	$\triangle_3$	...		$\triangle_i$	...	
k_0	k_1	k_2	k_3	...	k_{i-1}	$\rightleftarrows \mathrm{k}_i$	...	k_n
C_0	C_1	C_2	C_3	...	C_{i-1}	C_i	...	C_n

Bi-directional key updates:

- ✓ We can infer k_i from k_{i-1} and $\triangle_i$;
- ✓ We can infer k_{i-1} from k_i and $\triangle_i$;

Bi-directional Updates

0		1		2		3	...	i-1		i	...	n
	$\triangle_1$		$\triangle_2$		$\triangle_3$		...		$\triangle_i$		...	
k_0		k_1		k_2		k_3	...	k_{i-1}		k_i	...	k_n
C_0		C_1		C_2		C_3	...	C_{i-1}	$\rightleftarrows$	C_i	...	C_n

Bi-directional ciphertext updates:

- ✓ We can infer C_i from C_{i-1} and $\triangle_i$;
- ✓ We can infer C_{i-1} from C_i and $\triangle_i$;

(Forward-leak) Uni-directional Updates

0	1	2	3	...	i-1	i	...	n
$\triangle_1$	$\triangle_2$	$\triangle_3$		...	$\triangle_i$		...	
k_0	k_1	k_2	k_3	...	k_{i-1} $\underset{\not\Leftarrow}{\Rightarrow}$	k_i	...	k_n
C_0	C_1	C_2	C_3	...	C_{i-1}	C_i	...	C_n

Forward-leak Uni-directional key updates:

- ✓ We can only infer k_i from k_{i-1} and $\triangle_i$;
- ✗ We can not infer k_{i-1} from k_i and $\triangle_i$;

(Forward-leak) Uni-directional Updates

0		1		2		3	...	i-1		i	...	n
	$\triangle_1$		$\triangle_2$		$\triangle_3$		...		$\triangle_i$		...	
k_0		k_1		k_2		k_3	...	k_{i-1}		k_i	...	k_n
C_0		C_1		C_2		C_3	...	C_{i-1}	$\Rightarrow$ / $\not\Leftarrow$	C_i	...	C_n

(Forward-leak) uni-directional ciphertext updates:

- ✓ We can only infer C_i from C_{i-1} and $\triangle_i$;
- ✗ We can not infer C_{i-1} from C_i and $\triangle_i$;

(Backward-leak) Uni-directional Updates

0		1		2		3	...	i-1		i	...	n
	$\triangle_1$		$\triangle_2$		$\triangle_3$		...		$\triangle_i$		...	
k_0		k_1		k_2		k_3	...	k_{i-1}	$\not\rightarrow$ $\leftarrow$	k_i	...	k_n
C_0		C_1		C_2		C_3	...	C_{i-1}		C_i	...	C_n

Backward-leak Uni-directional key updates [Nishimaki, PKC'22]:

- ✓ We can only infer k_{i-1} from k_i and $\triangle_i$;
- ✗ We cannot infer k_i from k_{i-1} and $\triangle_i$;

(Forward-leak) uni-directional ciphertext updates:

- ✓ We can only infer C_i from C_{i-1} and $\triangle_i$;
- ✗ We cannot infer C_{i-1} from C_i and $\triangle_i$;

No-directional Key Updates

time $\longrightarrow$

0		1		2		3	...	i-1		i	...	n
	$\triangle_1$		$\triangle_2$		$\triangle_3$		...		$\triangle_i$		...	
k_0		k_1		k_2		k_3	...	k_{i-1}	$\not\rightleftarrows$	k_i	...	k_n
C_0		C_1		C_2		C_3	...	C_{i-1}		C_i	...	C_n

No-directional key updates:

- ✗ We cannot infer k_i from k_{i-1} and $\triangle_i$;
- ✗ We cannot infer k_{i-1} from k_i and $\triangle_i$;

Questions

- UE schemes with uni-directional updates leak less information than bi-directional updates
  - Are uni-directional updates more secure?

Questions

- UE schemes with uni-directional updates leak less information than bi-directional updates
  - Are uni-directional updates more secure?
- UE schemes with no-directional key updates leak the least information

Questions

- UE schemes with uni-directional updates leak less information than bi-directional updates
  - Are uni-directional updates more secure?
- UE schemes with no-directional key updates leak the least information
  - Are no-directional key updates most secure?

Does the Direction Matter?

- **[Jiang20] The direction of updatable encryption does not matter much**
  Jiang, Y.; ASIACRYPT 2020. (ePrint 2020/622)
- **[Nishimaki22] The direction of updatable encryption does matter.**
  Nishimaki, R.; PKC 2022 (ePrint 2021/221)

Does the Direction Matter?

- **[Jiang20] The direction of updatable encryption does not matter much**
  Jiang, Y.; ASIACRYPT 2020. (ePrint 2020/622)
- **[Nishimaki22] The direction of updatable encryption does matter.**
  Nishimaki, R.; PKC 2022 (ePrint 2021/221)

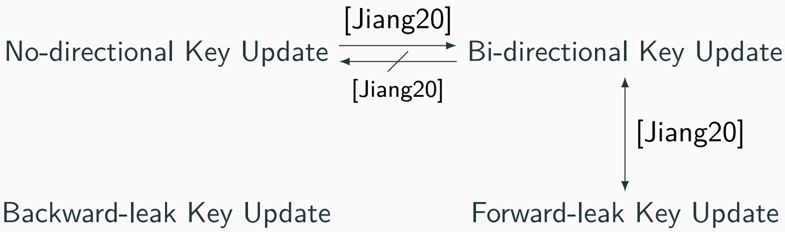
Contradiction?

A Closer Look

Setting: Confidentiality, (forward-leak) uni-directional CT updates

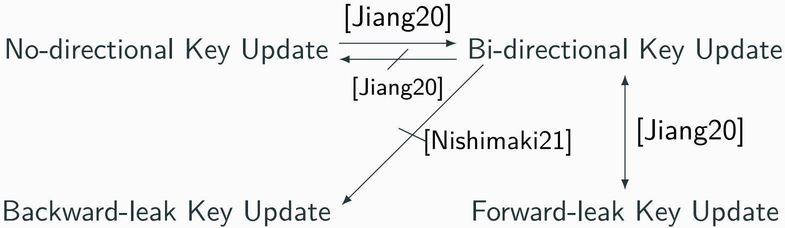
A Closer Look

Setting: Confidentiality, (forward-leak) uni-directional CT updates



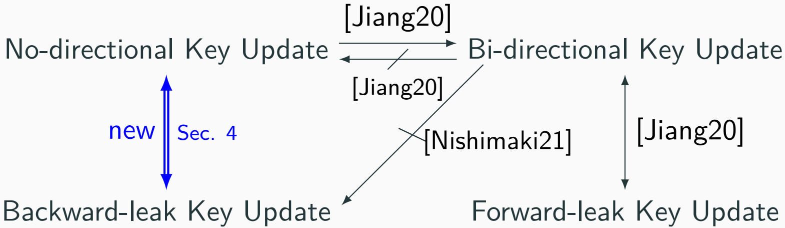
A Closer Look

Setting: Confidentiality, (forward-leak) uni-directional CT updates



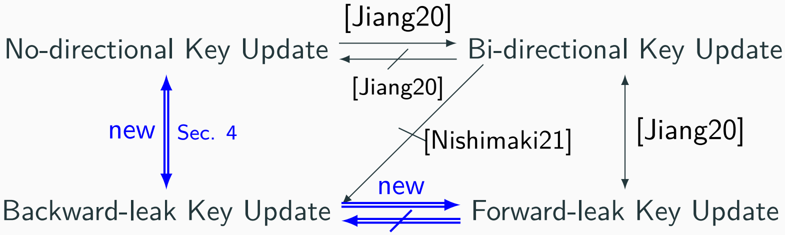
A Closer Look

Setting: Confidentiality, (forward-leak) uni-directional CT updates



A Closer Look

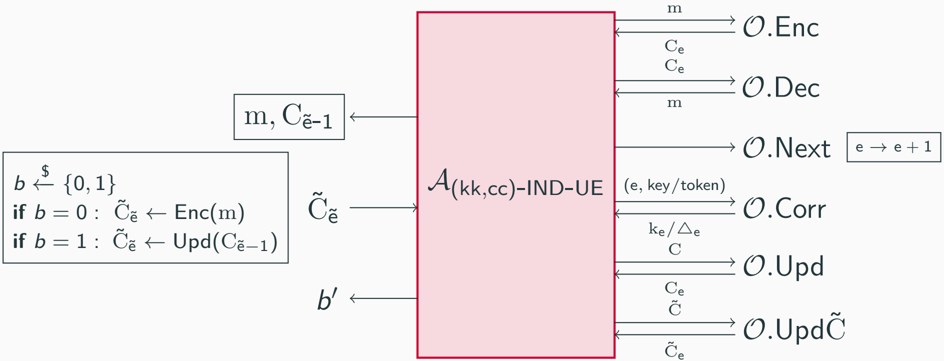
Setting: Confidentiality, (forward-leak) uni-directional CT updates



- **[Jiang20] The direction of updatable encryption does not matter much**
  Jiang; ASIACRYPT 2020. (ePrint 2020/622)
- **[Nishimaki22] The direction of updatable encryption does matter.**
  Nishimaki; PKC 2022 (ePrint 2021/221)

Confidentiality Notions

- For kk $\in$ {f-uni, b-uni, bi, no} and cc $\in$ {uni, bi}, consider UE schemes with kk-directional key updates and cc to cc-directional ciphertext updates.



- Challenger checks leaked information to see if the adversary can trivially win
- Trivial wins depend on the update settings!

How to prove these relations?

- To prove:
  - Backward-leak $\Leftrightarrow$ No-directional
- Proof idea:
  - The trivial wins in the backward-uni-directional updates are triggered if and only if the trivial wins in the no-directional updates are triggered.

Generic Constructions of UE

Prior Work: Direct Constructions from Concrete Assumptions

UE schemes	Assumptions	IND-UE
[Nishimaki22]	LWE	backward-leak
[Nishimaki22]	IO	no-directional
RISE [LT18] and SHINE [BDGJ20]	DDH	bi-directional
Encrypt-and-MAC (E&M) [KLR19]	DDH + ROM	bi-directional
NYUAE [KLR19]	SXDH	bi-directional
LWEUE [Jiang20]	LWE	bi-directional

Prior Work: Direct Constructions from Concrete Assumptions

UE schemes	Assumptions	IND-UE
[Nishimaki22]	LWE	backward-leak
[Nishimaki22]	IO	no-directional
RISE [LT18] and SHINE [BDGJ20]	DDH	bi-directional
Encrypt-and-MAC (E&M) [KLR19]	DDH + ROM	bi-directional
NYUAE [KLR19]	SXDH	bi-directional
LWEUE [Jiang20]	LWE	bi-directional

- Backward-leak or no-directional UE schemes from Other Assumptions?

Prior Work: Direct Constructions from Concrete Assumptions

UE schemes	Assumptions	IND-UE
[Nishimaki22]	LWE	backward-leak
[Nishimaki22]	IO	no-directional
RISE [LT18] and SHINE [BDGJ20]	DDH	bi-directional
Encrypt-and-MAC (E&M) [KLR19]	DDH + ROM	bi-directional
NYUAE [KLR19]	SXDH	bi-directional
LWEUE [Jiang20]	LWE	bi-directional

- Backward-leak or no-directional UE schemes from Other Assumptions?
- Generic constructions of UE?

Prior Work: Direct Constructions from Concrete Assumptions

UE schemes	Assumptions	IND-UE
[Nishimaki22]	LWE	backward-leak
[Nishimaki22]	IO	no-directional
RISE [LT18] and SHINE [BDGJ20]	DDH	bi-directional
Encrypt-and-MAC (E&M) [KLR19]	DDH + ROM	bi-directional
NYUAE [KLR19]	SXDH	bi-directional
LWEUE [Jiang20]	LWE	bi-directional

- Backward-leak or no-directional UE schemes from Other Assumptions?
- Generic constructions of UE?

Key and Message Homomorphic PKE

- Key Homomorphic PKE:
  - the distribution generated from the homomorphism of keys is statistically close to the original key distribution
  - allow to compute a new public key from a secret key (assume sk_2) and an old public key (assume pk_1 with sk_1 as its secret key), then

$$pk_{new} = KHK(sk_2, pk_1) \overset{s}{\approx} [sk_1 \otimes sk_2],$$

- Message Homomorphic PKE: for any message $m_1, m_2 \in M$ and any public key pk:

$$Enc(pk, m_1) \otimes Enc(pk, m_2) = Enc(pk, m_1 \oplus m_2)$$

Examples
The ElGamal and Regev encryption schemes

Uni-Directional UE from Key and Message Homomorphic PKE

<u>Setup(λ) :</u>

$(sk_{1,1}, pk_{1,1}) \leftarrow \mathsf{PKE.KG}(\lambda)$
return $(sk_{1,1}, pk_{1,1})$

<u>Next($\mathbf{sk}_e$) :</u>

parse $\mathbf{sk}_e = (sk_{e,1}, \ldots, sk_{e,e})$
for $i \in \{1, \ldots, e\}$ **do**
  $(\Delta_i, [\Delta_i]) \leftarrow \mathsf{PKE.KG}(\lambda)$
  $sk_{e+1,i} \leftarrow sk_{e,i} \otimes \Delta_i$
  $pk_{e+1,i} \leftarrow [sk_{e+1,i}]$
$(sk_{e+1,e+1}, pk_{e+1,e+1}) \leftarrow \mathsf{PKE.KG}(\lambda)$
$\mathbf{sk}_{e+1} \leftarrow (sk_{e+1,1}, \ldots, sk_{e+1,e+1})$
$\mathbf{pk}_{e+1} \leftarrow (pk_{e+1,1}, \ldots, pk_{e+1,e+1})$
$\mathbf{\Delta}^{sk}_{e+1} \leftarrow (\Delta_1, \ldots, \Delta_e)$
$\mathbf{\Delta}_{e+1} \leftarrow (\mathbf{\Delta}^{sk}_{e+1}, pk_{e+1,e+1})$
return $\mathbf{\Delta}_{e+1}, (\mathbf{sk}_{e+1}, \mathbf{pk}_{e+1})$

<u>Enc($\mathbf{pk}_e, m$) :</u>

$\mathbf{R}_e \xleftarrow{\$} \mathcal{M}^{e \times 1}$
parse $\mathbf{R}_e = (r_{e,1}, \ldots, r_{e,e})$
$\mathbf{c}_{e,1} \leftarrow \mathsf{PKE.Enc}(\mathbf{pk}_e, \mathbf{R}_e)$
$c_{e,2} \leftarrow r_{e,1} \oplus \cdots \oplus r_{e,e} \oplus m$
return $\mathbf{c}_e = (\mathbf{c}_{e,1}, c_{e,2})$

<u>Dec($\mathbf{sk}_e, \mathbf{c}_e$) :</u>

parse $\mathbf{c}_e = (\mathbf{c}_{e,1}, c_{e,2})$
$\mathbf{R}_e \leftarrow \mathsf{PKE.Dec}(\mathbf{sk}_e, \mathbf{c}_{e,1})$
parse $\mathbf{R}_e = (r_{e,1}, \ldots, r_{e,e})$
$m' \leftarrow c_{e,2} \oplus^{-1} (r_{e,1} \oplus \cdots \oplus r_{e,e})$
return m'

<u>Upd($\mathbf{\Delta}_{e+1}, \mathbf{c}_e$) :</u>

parse $\mathbf{\Delta}_{e+1} = (\mathbf{\Delta}^{sk}_{e+1}, pk_{e+1})$
parse $\mathbf{c}_e = (\mathbf{c}_{e,1}, c_{e,2})$
$\mathbf{R} \xleftarrow{\$} \mathcal{M}^{(e+1) \times 1}$
$\mathbf{c}^1 \leftarrow \mathsf{PKE.KHC}(\mathbf{\Delta}^{sk}_{e+1}, \mathbf{c}_{e,1})$
$\mathbf{c}_{e+1,1} \leftarrow (\mathbf{c}^1, 0) + \mathsf{PKE.Enc}(\mathbf{pk}_{e+1}, \mathbf{R})$
parse $\mathbf{R} = (r_1, \ldots, r_{e+1})$
$c_{e+1,2} \leftarrow c_{e,2} \oplus r_1 \oplus \cdots \oplus r_{e+1}$
$\mathbf{c}_{e+1} \leftarrow (\mathbf{c}_{e+1,1}, c_{e+1,2})$
return $\mathbf{c}_{e+1}$

Bootstrappable PKE

- $\exists Recrypt$ s.t. $\forall (sk_1, pk_1), (sk_2, pk_2) \xleftarrow{\$} KG(\lambda)$ and m

$$(c, Recrypt(pk_2, D, Enc(pk_2, sk_1), c)) \overset{s}{\approx} (c, Enc(pk_2, m))$$

  where $c = Enc(pk_1, m)$ and D is its own decryption circuit.

Bootstrappable PKE

- $\exists \mathit{Recrypt}$ s.t. $\forall (sk_1, pk_1), (sk_2, pk_2) \xleftarrow{\$} KG(\lambda)$ and m

$$(c, \mathit{Recrypt}(pk_2, D, \mathit{Enc}(pk_2, sk_1), c)) \overset{s}{\approx} (c, \mathit{Enc}(pk_2, m))$$

  where $c = \mathit{Enc}(pk_1, m)$ and D is its own decryption circuit.
- *Recrypt*: For updating CT

UE from Bootstrappable PKE

$\underline{\mathsf{Setup}(\lambda):}$

$(sk_1, pk_1) \leftarrow \mathsf{BPKE.KG}(\lambda)$

return (sk_1, pk_1)

$\underline{\mathsf{Next}(sk_\mathsf{e}):}$

$(sk_{\mathsf{e}+1}, pk_{\mathsf{e}+1}) \leftarrow \mathsf{BPKE.KG}(\lambda)$

$\Delta_{\mathsf{e}+1} \leftarrow \mathsf{BPKE.Enc}(pk_{\mathsf{e}+1}, sk_\mathsf{e})$

return $\Delta_{\mathsf{e}+1}, (sk_{\mathsf{e}+1}, pk_{\mathsf{e}+1})$

$\underline{\mathsf{Enc}(pk_\mathsf{e}, m):}$

$c_\mathsf{e} \leftarrow \mathsf{BPKE.Enc}(pk_\mathsf{e}, m)$

return c_e

$\underline{\mathsf{Dec}(sk_\mathsf{e}, c_\mathsf{e}):}$

$m' \leftarrow \mathsf{BPKE.Dec}(sk_\mathsf{e}, c_\mathsf{e})$

return m'

$\underline{\mathsf{Upd}(\Delta_{\mathsf{e}+1}, c_\mathsf{e}):}$

$c_{\mathsf{e}+1} \leftarrow \mathsf{BPKE.Recrypt}(pk_{\mathsf{e}+1}, D, \Delta_{\mathsf{e}+1}, c_\mathsf{e})$

return $c_{\mathsf{e}+1}$

Summary

Summary

- Equivalence: Under uni-directional ciphertext updates,

$$\text{Backward-leak uni-directional key update} \Leftrightarrow \text{No-directional}$$

- Two Generic Constructions of backward-leak UE:
  - From homomorphic PKE (concurrent to Miao, Patranabis, Watson, PKC 2023)
  - From bootstrappable PKE

More Efficient Construction?

- Uni-directional UE from standard assumptions without linear growth in the key and cipehrtextext

References

- **[LT18] Updatable encryption with post-compromise security**
  Lehmann, and Tackmann; Eurocrypt 2018. (ePrint 2018/118)
- **[KLR19] (R)CCA Secure Updatable Encryption with Integrity Protection**
  Klooß, Lehmann, and Rupp; Eurocrypt 2019. (ePrint 2019/222)
- **[BDGJ20] Fast and Secure Updatable Encryption**
  Boyd, Davies, Gjøsteen, and Jiang; Crypto 2020. (ePrint 2019/1457)
- **[Jiang20] The direction of updatable encryption does not matter much**
  Jiang; Asiacrypt 2020. (ePrint 2020/622)
- **[Nishimaki22] The direction of updatable encryption does matter.**
  Nishimaki; PKC 2022 (ePrint 2021/221)

Thank you for your attention!

Questions?