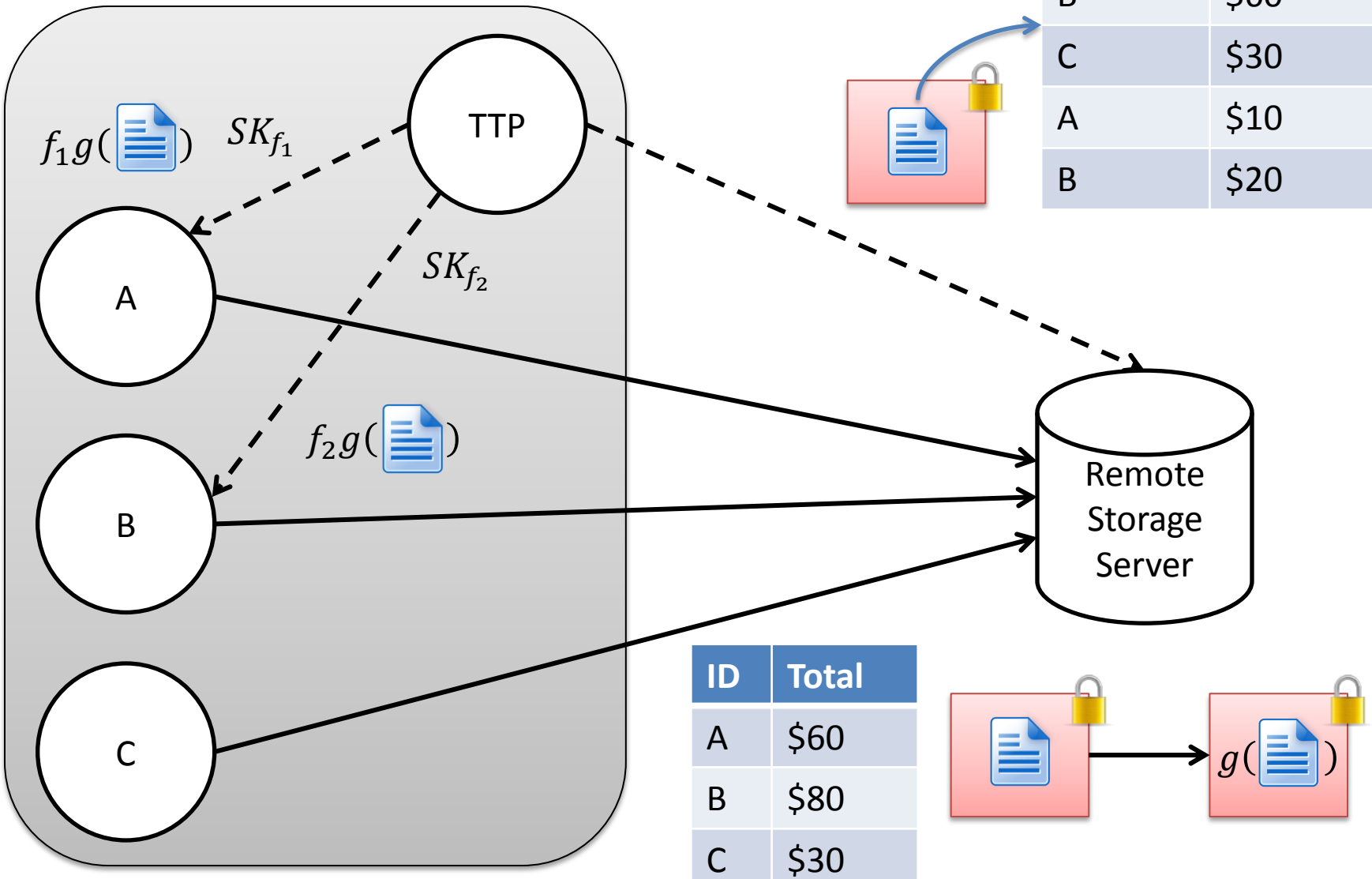
(The journey towards)

Functional Controllable Homomorphic Encryption

Ameer Mohammed
Soheil Nemati

Motivation



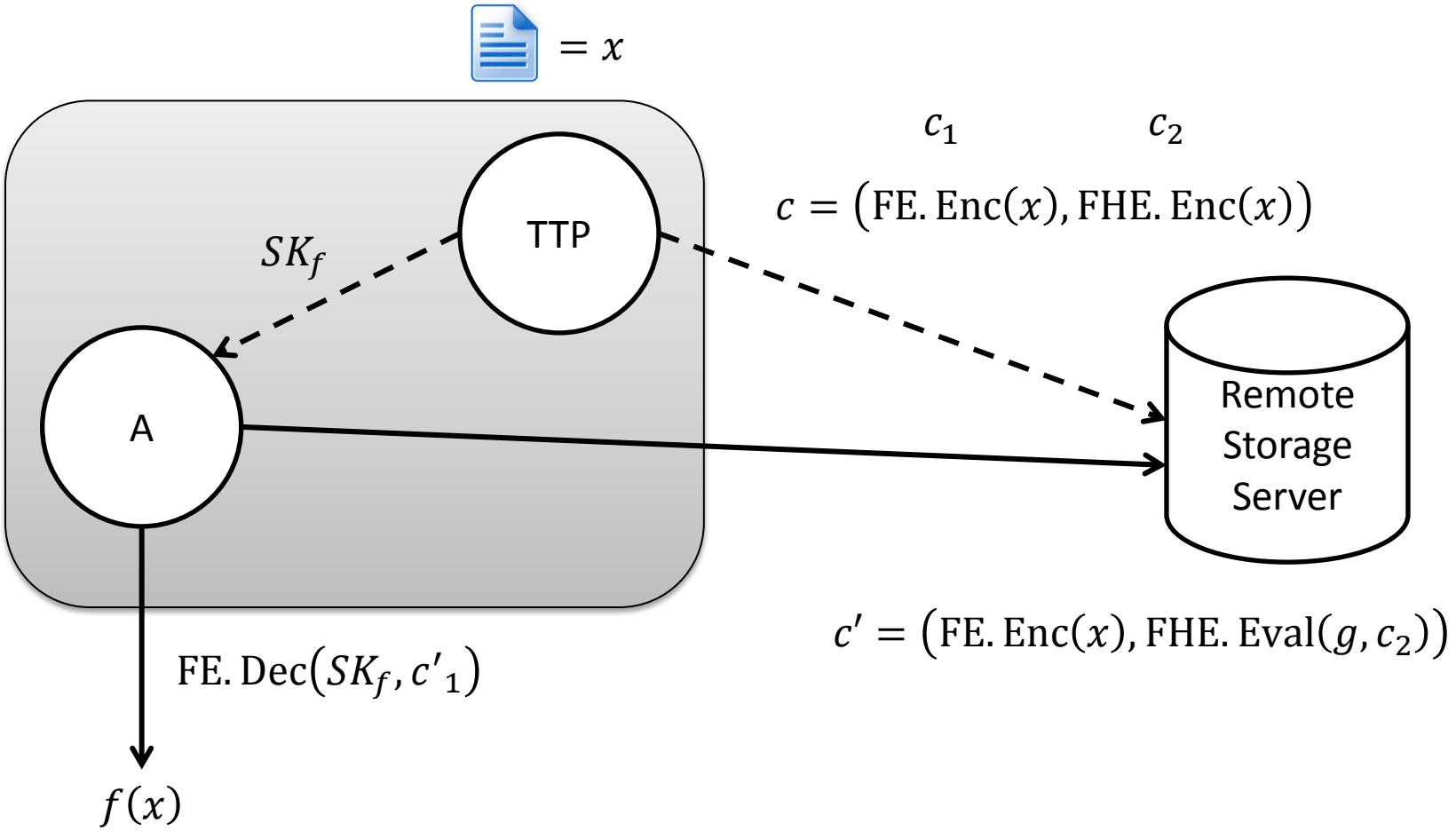
ID	Income
A	$50
B	$60
C	$30
A	$10
B	$20

ID	Total
A	$60
B	$80
C	$30

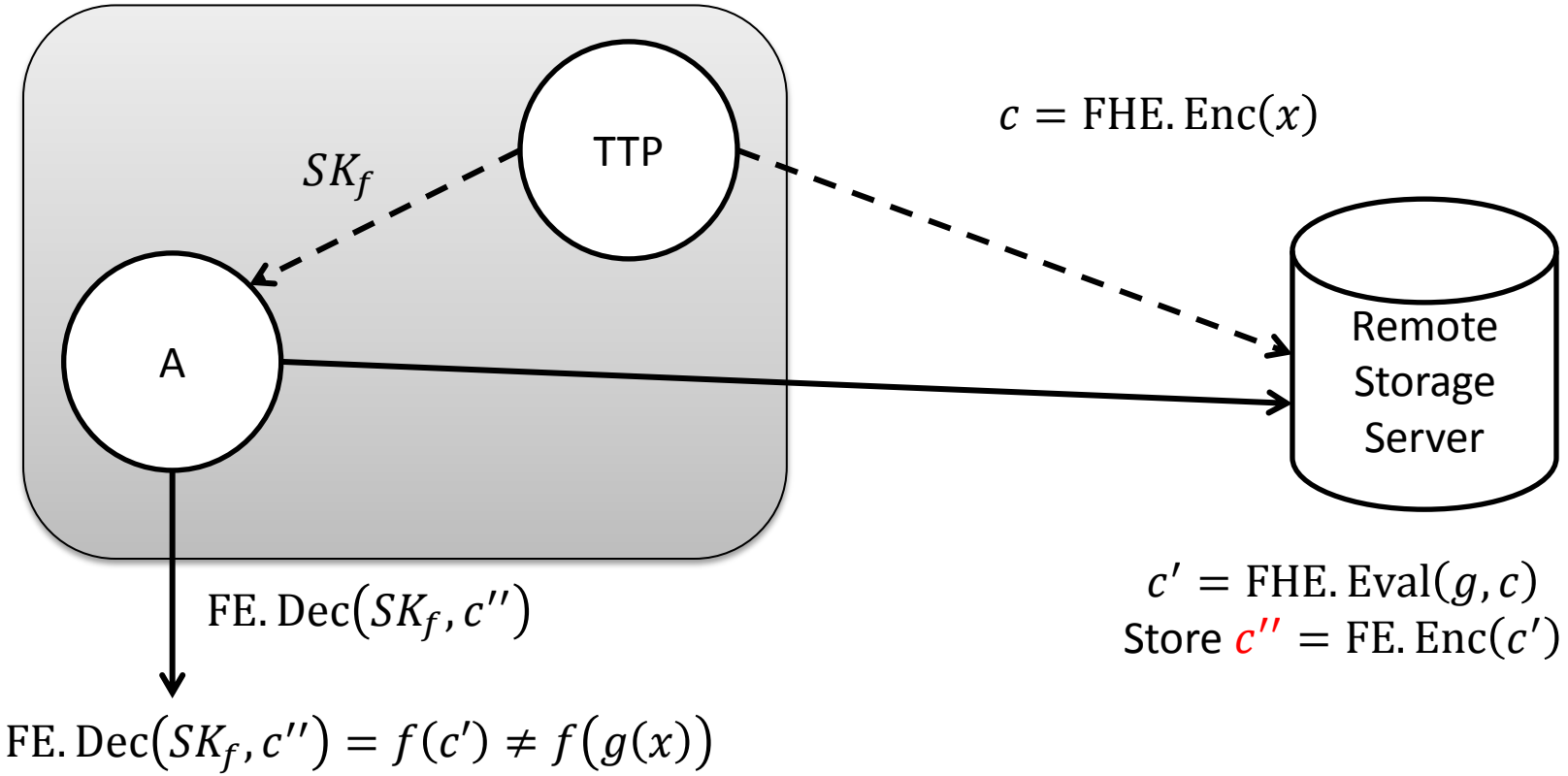
Refresher:
FE vs. FHE

Functional Encryption	Fully Homomorphic Encryption
$(PK, MSK) \leftarrow \text{Setup}(1^\kappa)$	$(PK, SK) \leftarrow \text{Setup}(1^\kappa)$
$SK_f \leftarrow \text{Keygen}(MSK, f)$	
$c \leftarrow \text{Enc}(MPK, x)$	$c \leftarrow \text{Enc}(MPK, x)$
	$c' \leftarrow \text{Eval}(PK, g, c_1, \dots, c_n)$
$f(x) \leftarrow \text{Dec}(SK_f, c)$	$g(x_1, \dots, x_n) \leftarrow \text{Dec}(SK, c')$

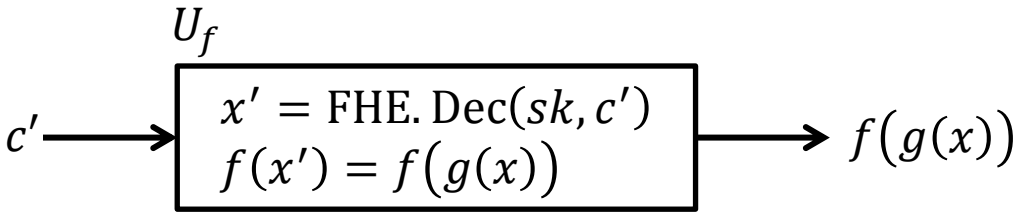
1st Try: Using FE + FHE

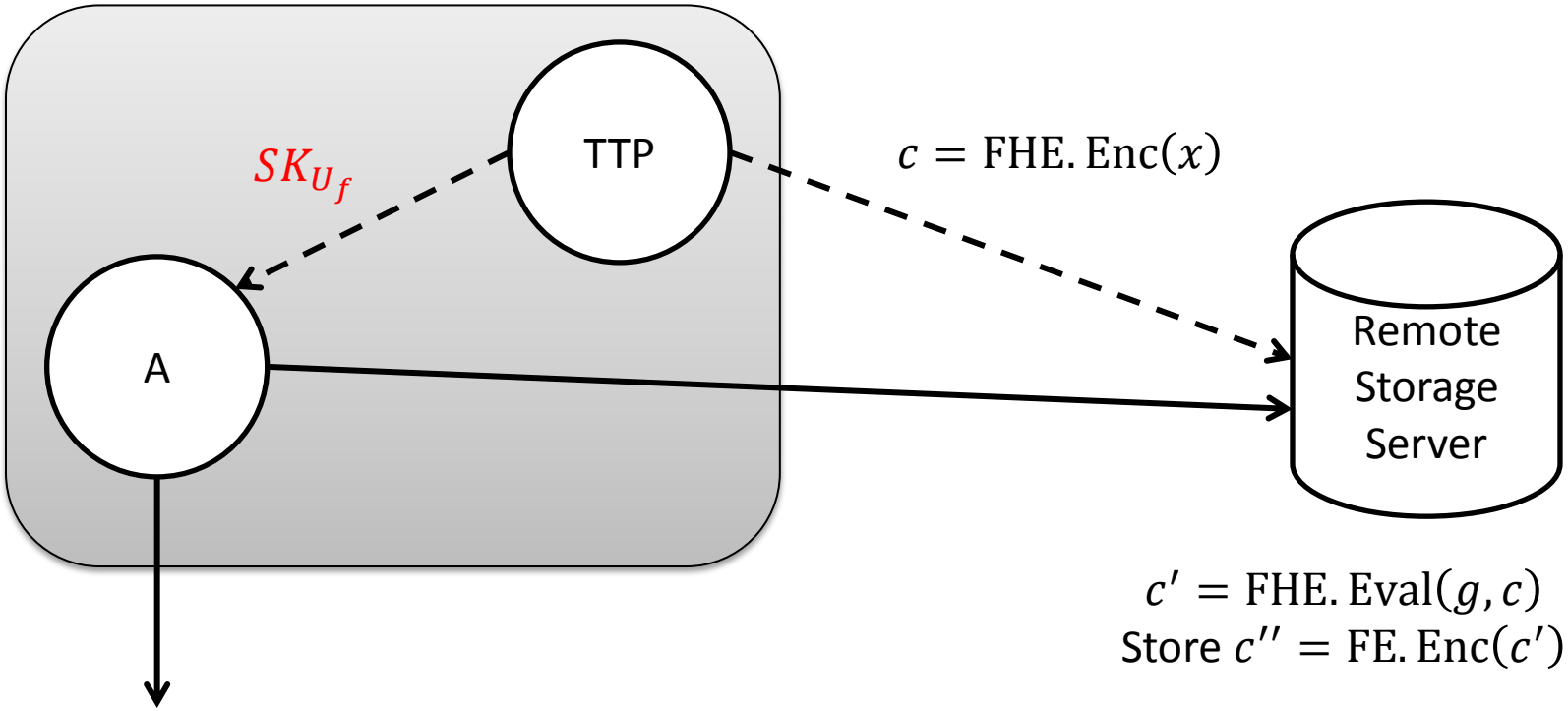


2nd Try: Using FE(FHE)

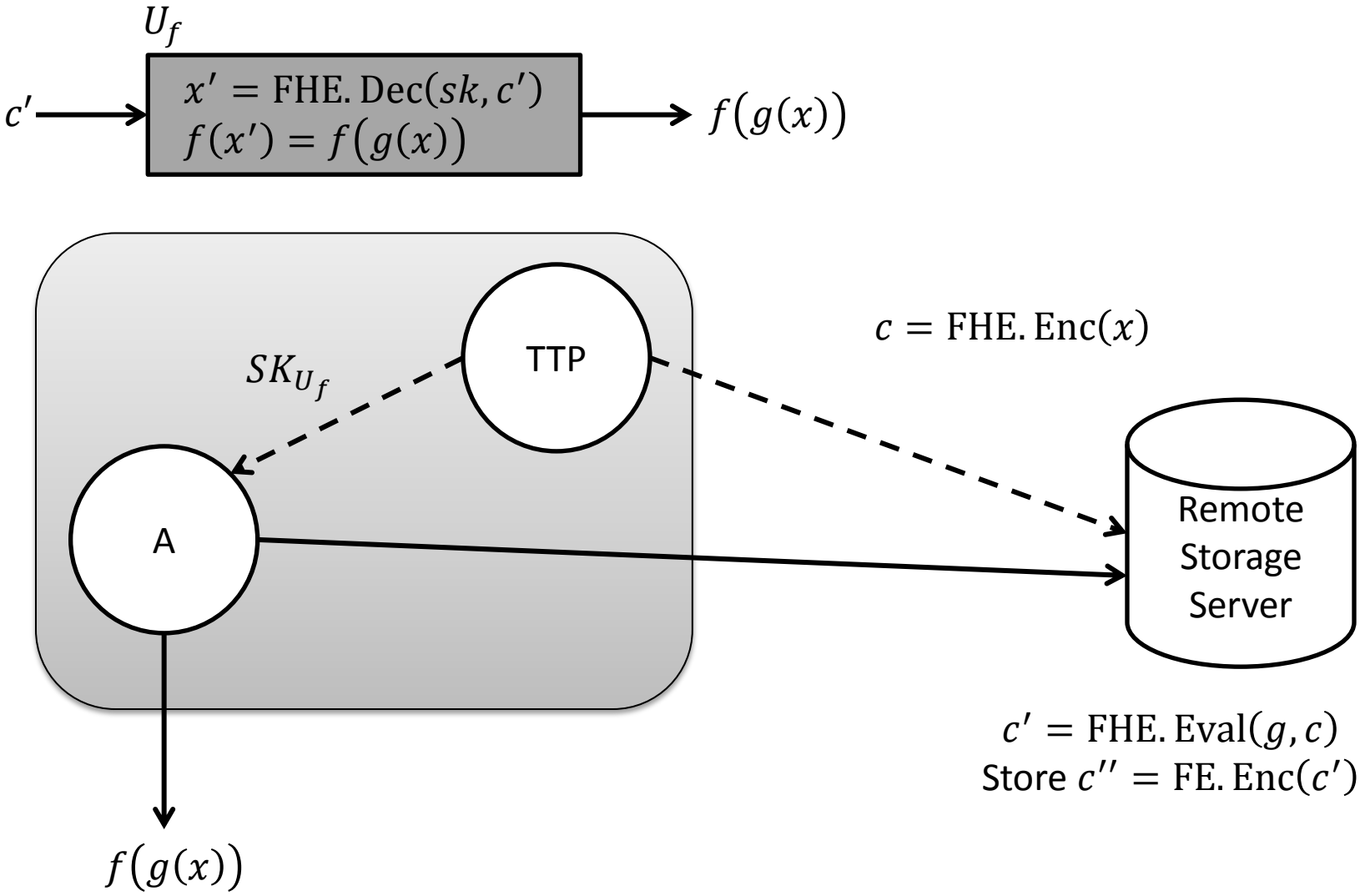


3rd Try: Using FE(FHE)



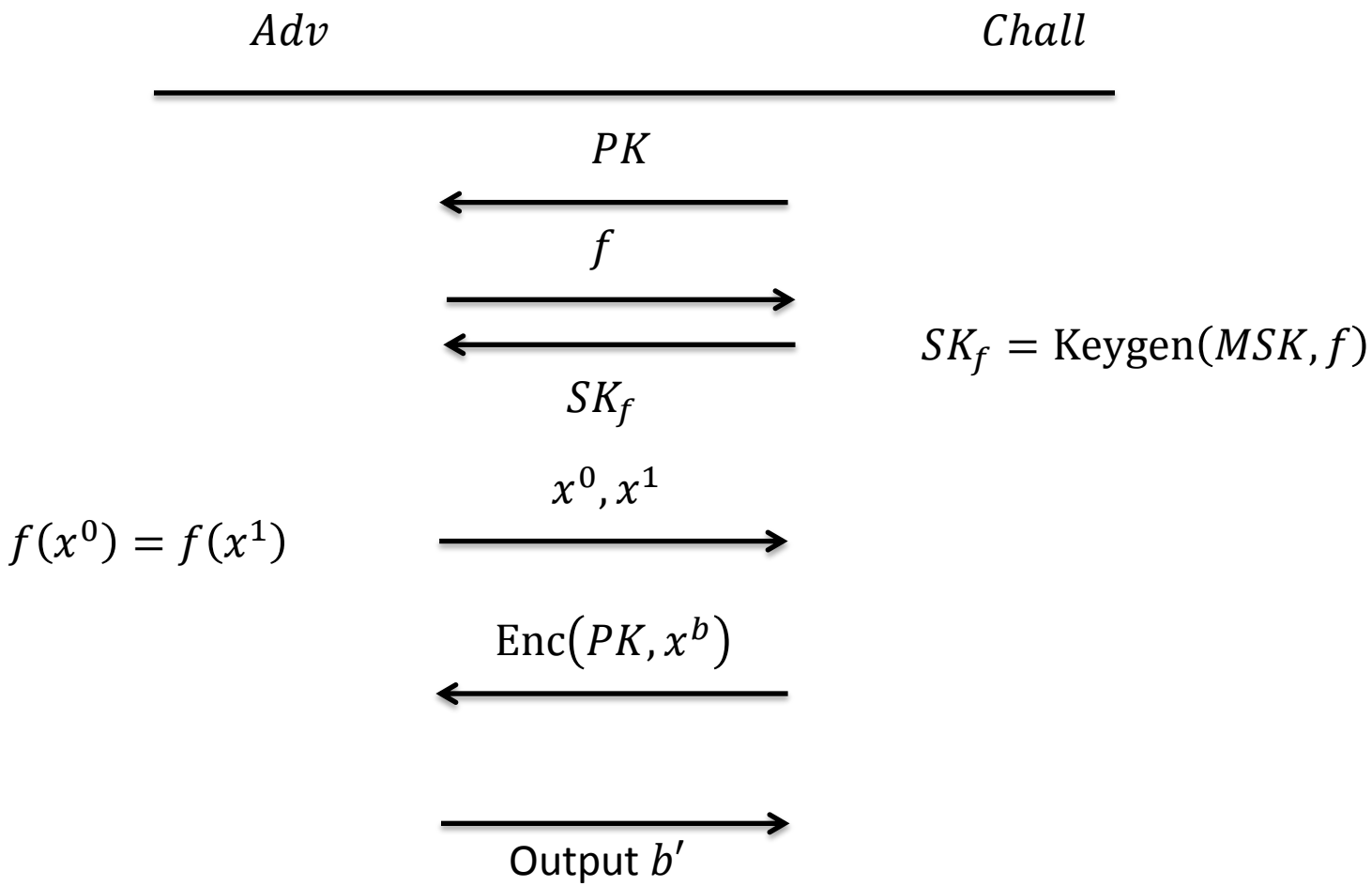


Next Step?


U_f
c'
$x' = \mathrm{FHE.Dec}(sk, c')$
$f(x') = f(g(x))$
$f(g(x))$
SK_{U_f}
TTP
$c = \mathrm{FHE.Enc}(x)$
A
Remote Storage Server
$c' = \mathrm{FHE.Eval}(g, c)$
Store $c'' = \mathrm{FE.Enc}(c')$
$f(g(x))$

Can we ***even*** achieve Functional Encryption that is also fully homomorphic?

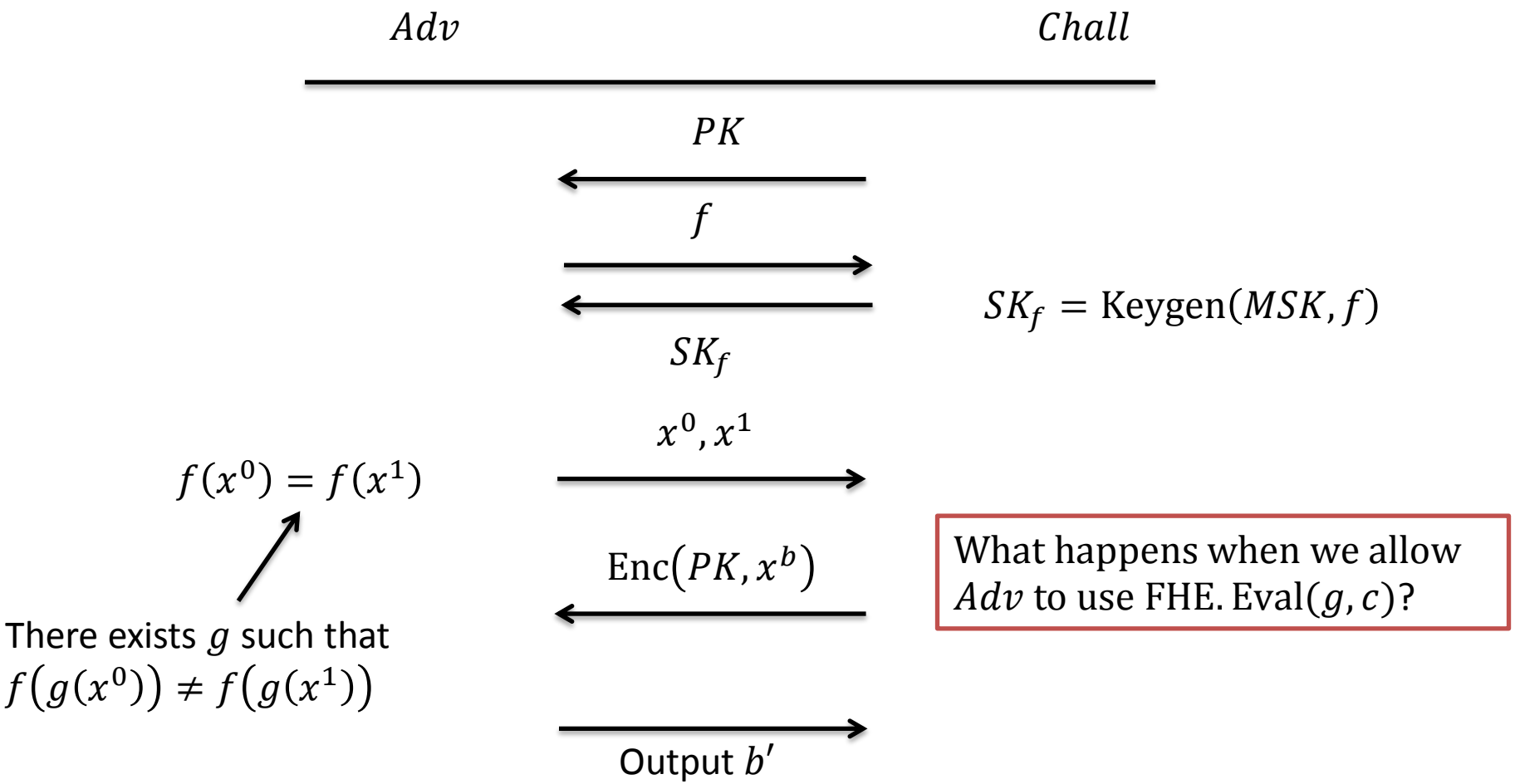
Functional Encryption Security Game



Secure if

$\Pr[b' = b] \leq \frac{1}{2} + negl(\kappa)$

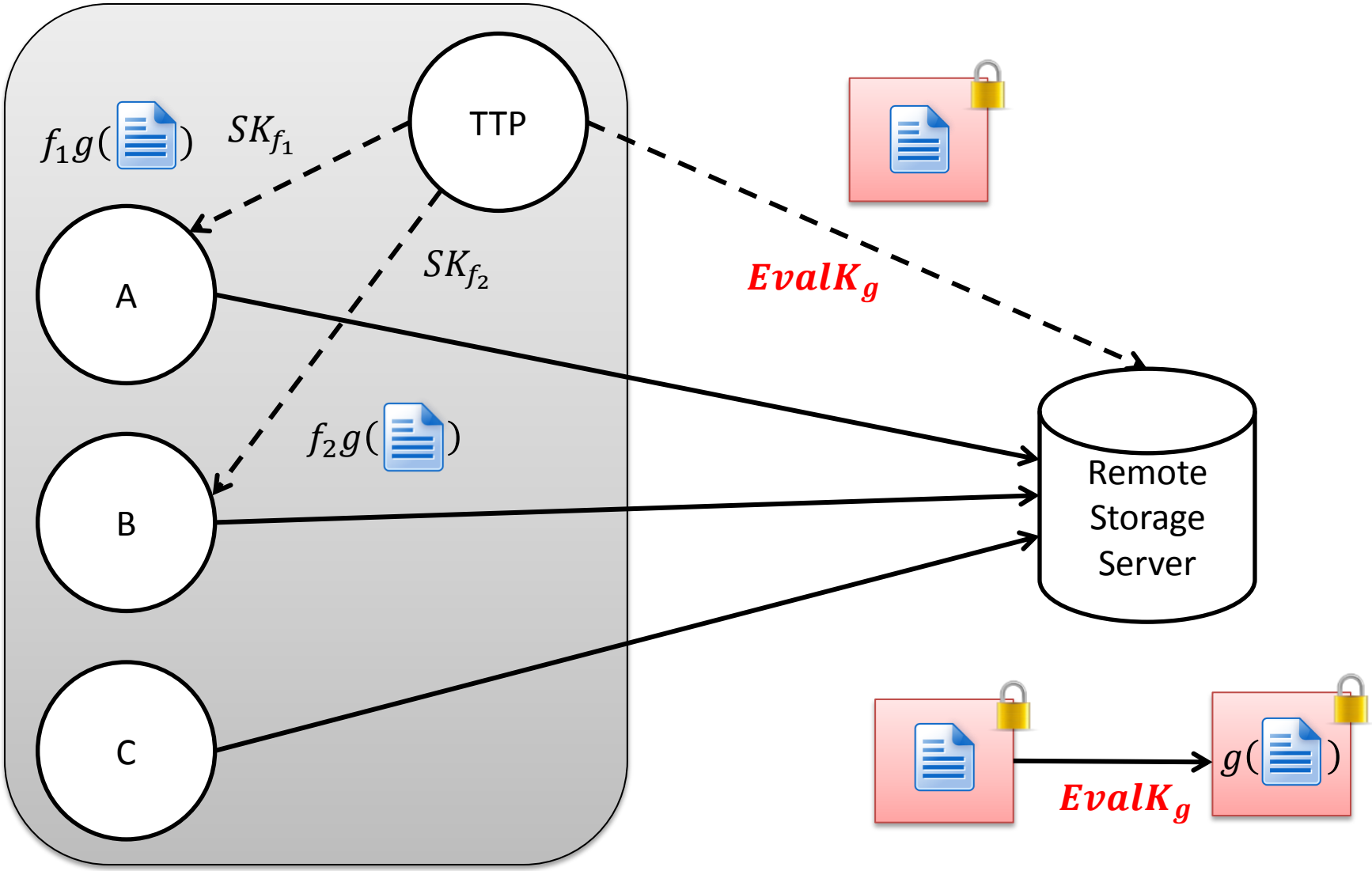
Func. Fully HE Security Game



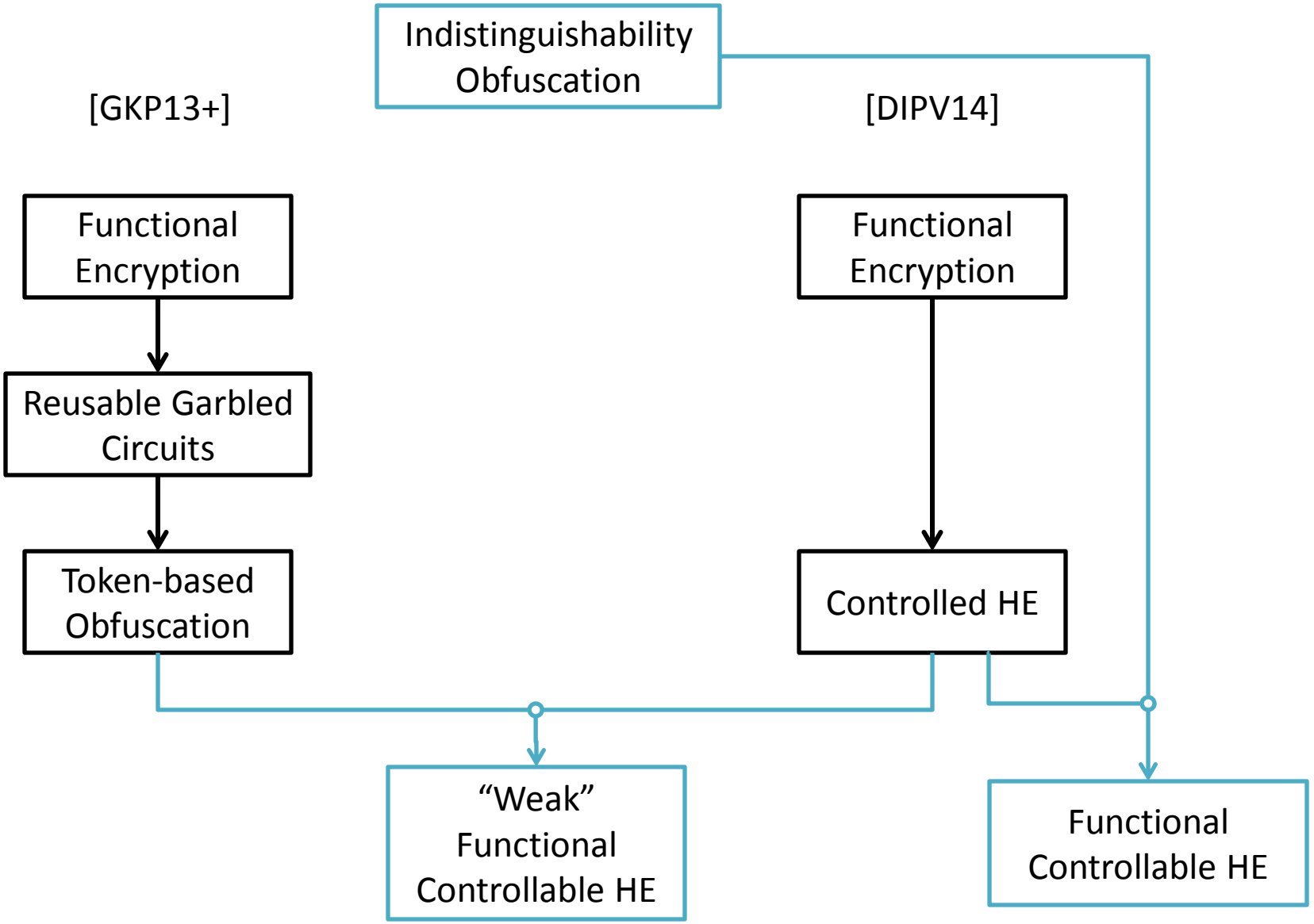
Secure if

$$\Pr[b' = b] \leq \frac{1}{2} + negl(\kappa)$$

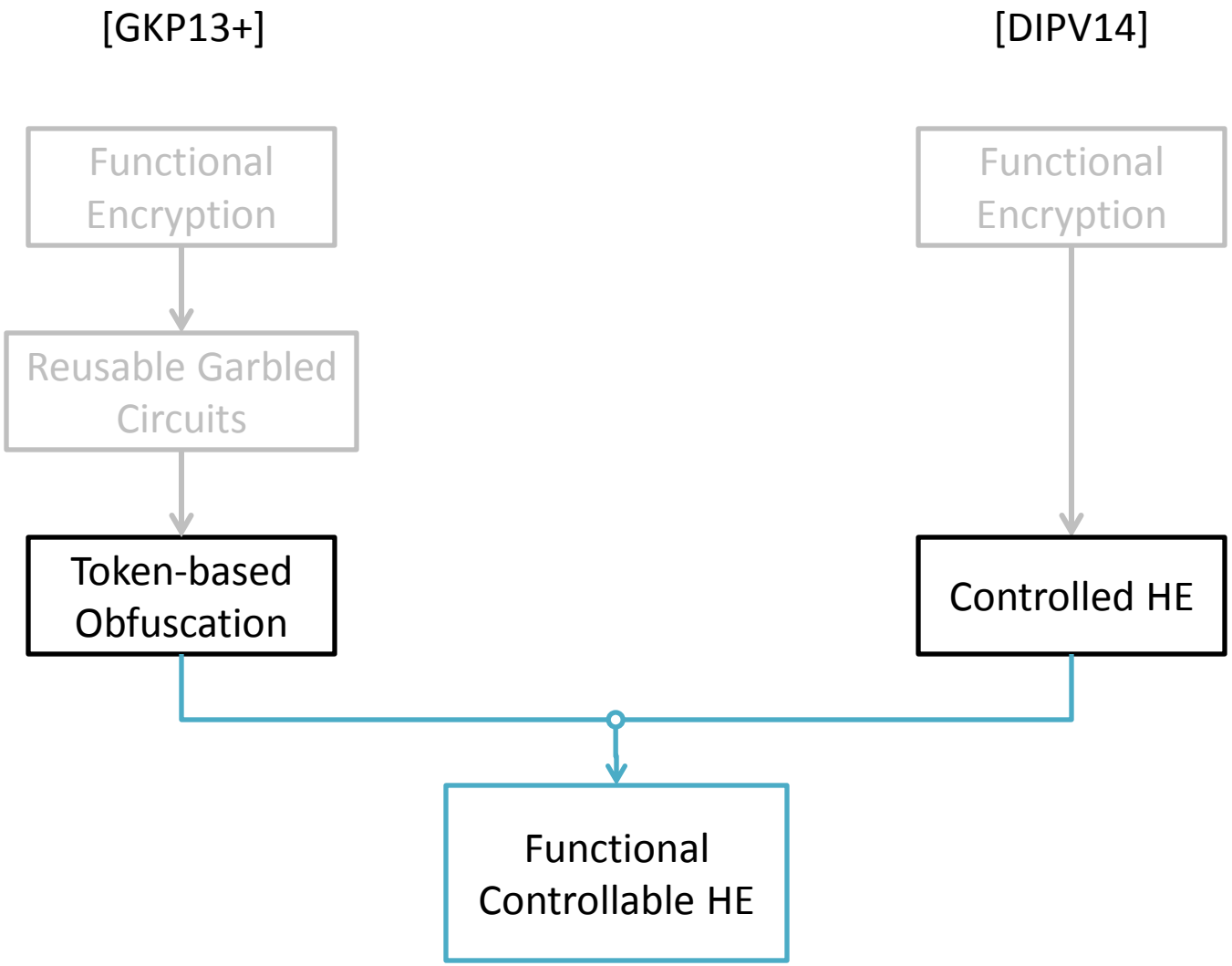
Motivation (Revisited)



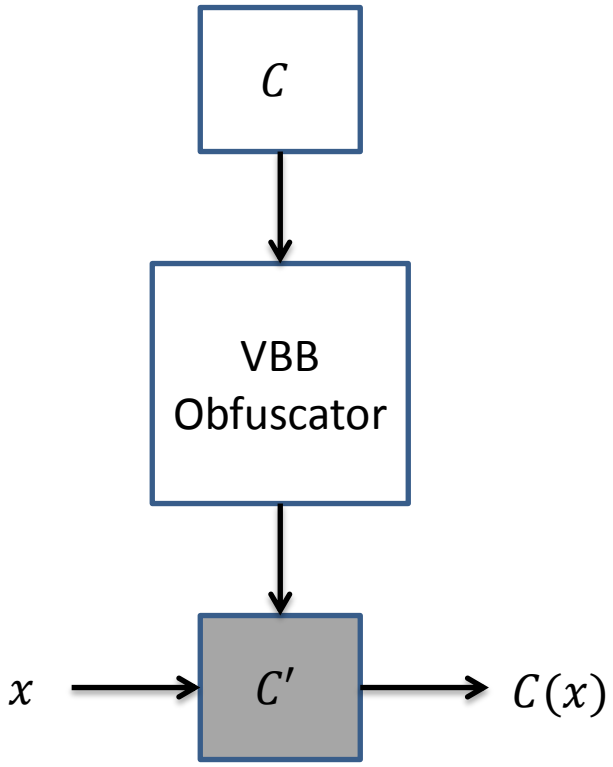
Overview of Techniques



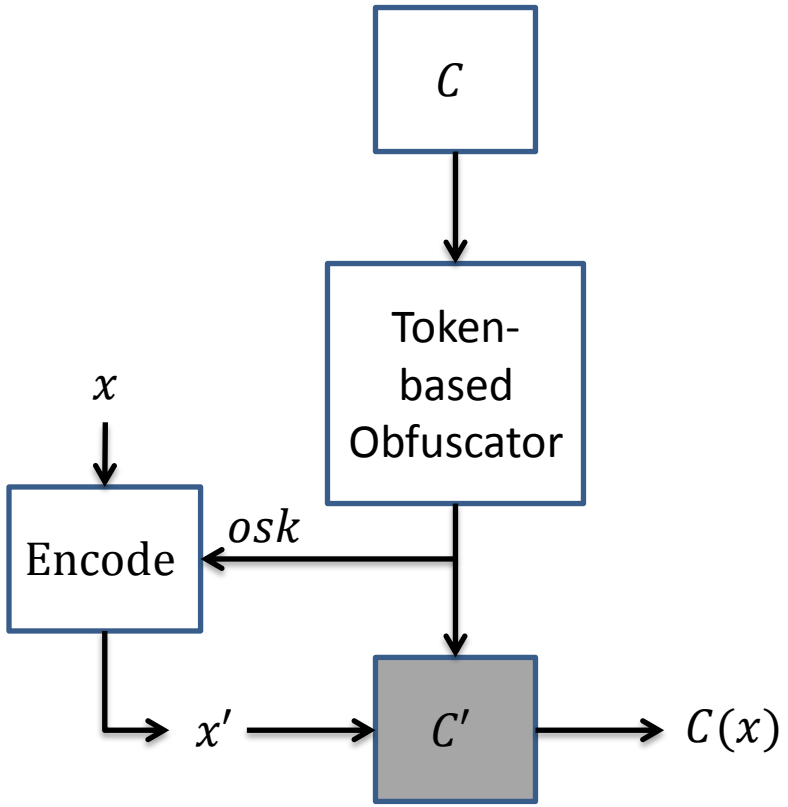
Func. Cont. HE


[GKP13+]
[DIPV14]
Functional Encryption
Reusable Garbled Circuits
Token-based Obfuscation
Functional Encryption
Controlled HE
Functional Controllable HE

What is Token-based Obfuscation?



Impossible [BGI+01]

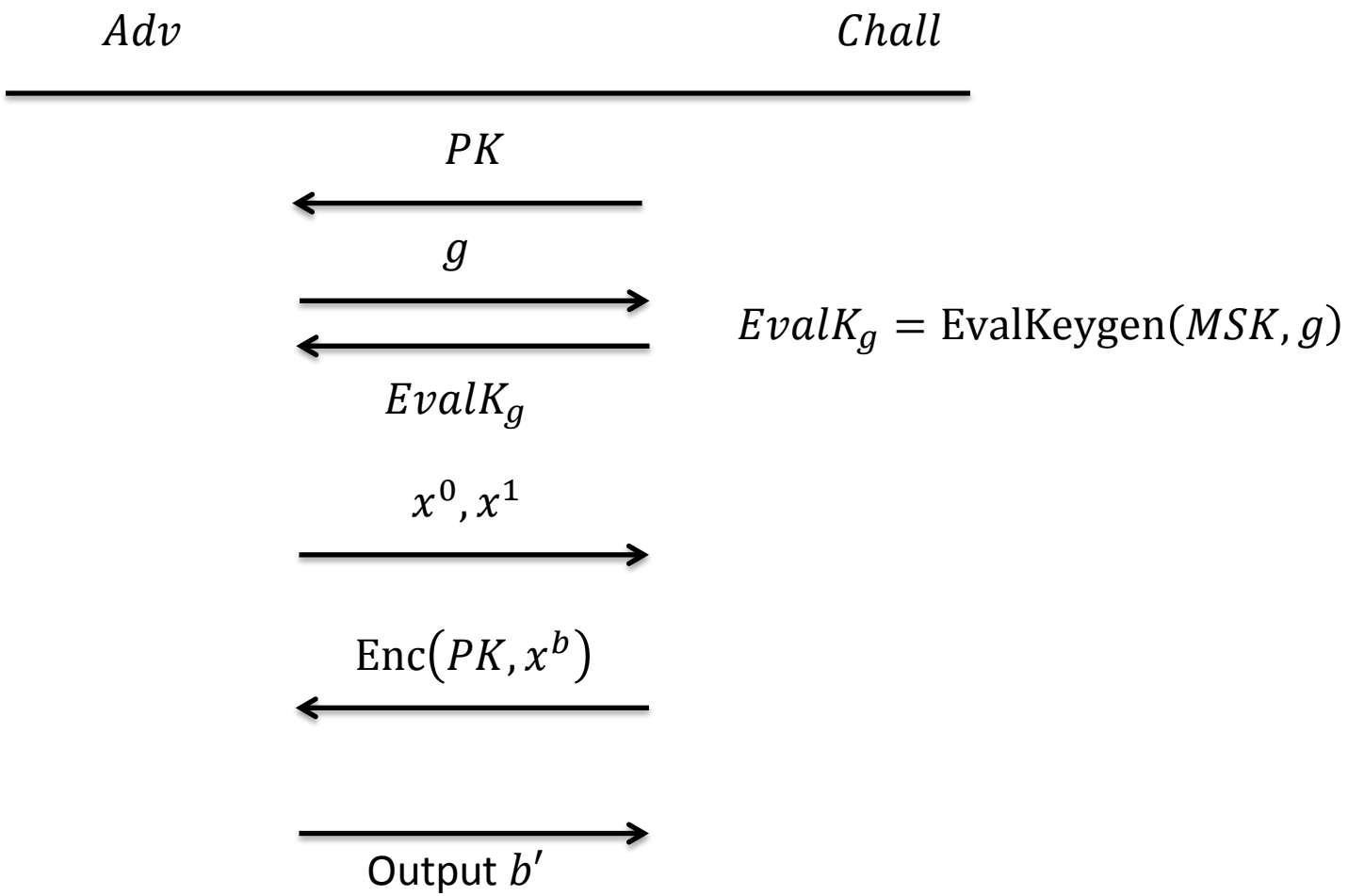


Possible [GKP+13]

What is Controlled Homomorphic Encryption?

Fully Homomorphic Encryption	Controlled Homomorphic Encryption
$(PK, SK) \leftarrow \mathrm{Setup}(1^n)$	$(MSK, PK) \leftarrow \mathrm{Setup}(1^n)$
$c \leftarrow \mathrm{Enc}(PK, x)$	$c \leftarrow \mathrm{Enc}(PK, x)$
	$EvalK_g \leftarrow \mathrm{EvalKeygen}(MSK, g)$
$c' \leftarrow \mathrm{Eval}(PK, g, c)$	$c' \leftarrow \mathrm{HEval}(PK, EvalK_g, c)$
$g(x) \leftarrow \mathrm{Dec}(SK, c')$	$g(x) \leftarrow \mathrm{Dec}(MSK, c')$

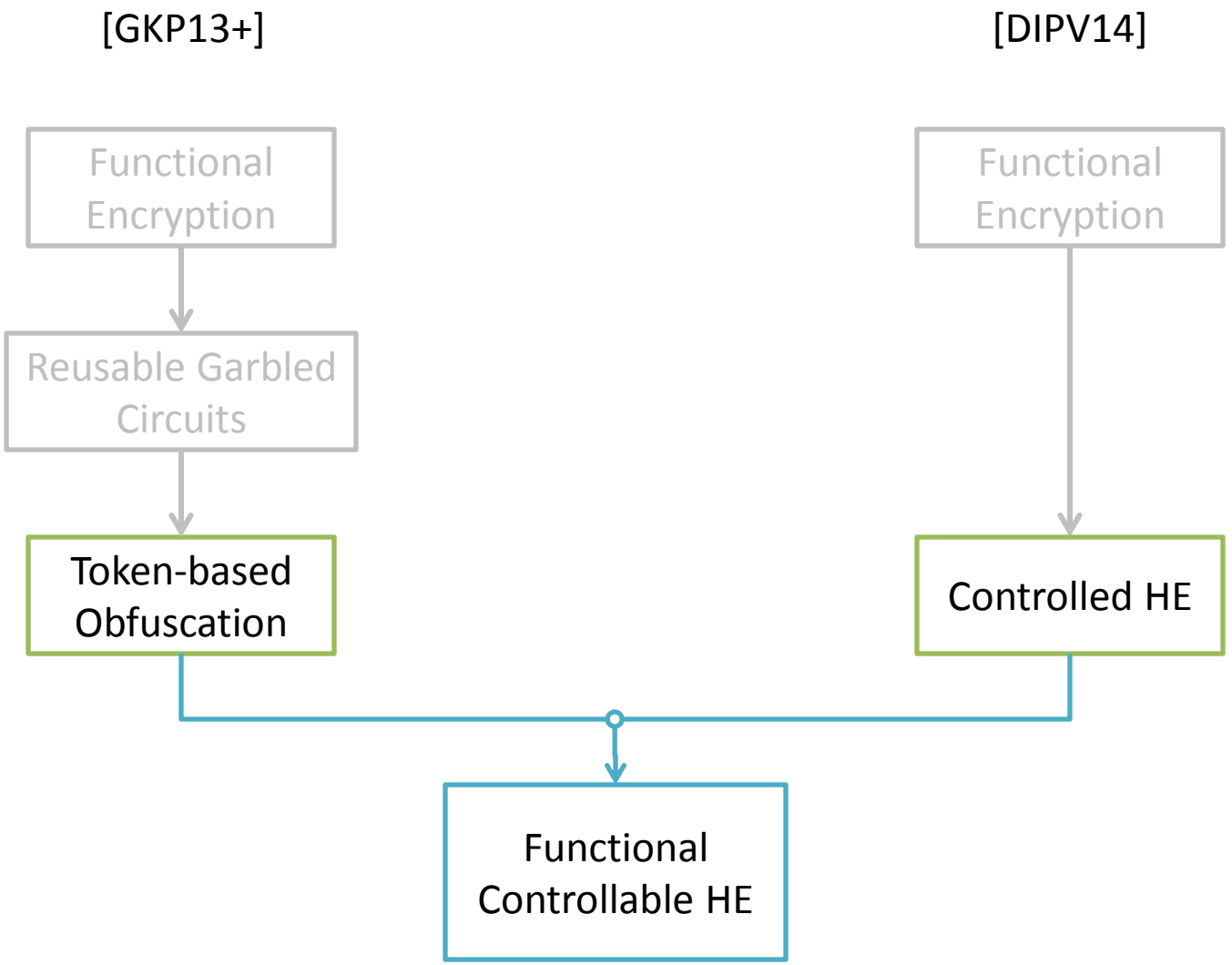
Controlled HE Security Game



Secure if

$$\Pr[b' = b] \leq \frac{1}{2} + negl(\kappa)$$

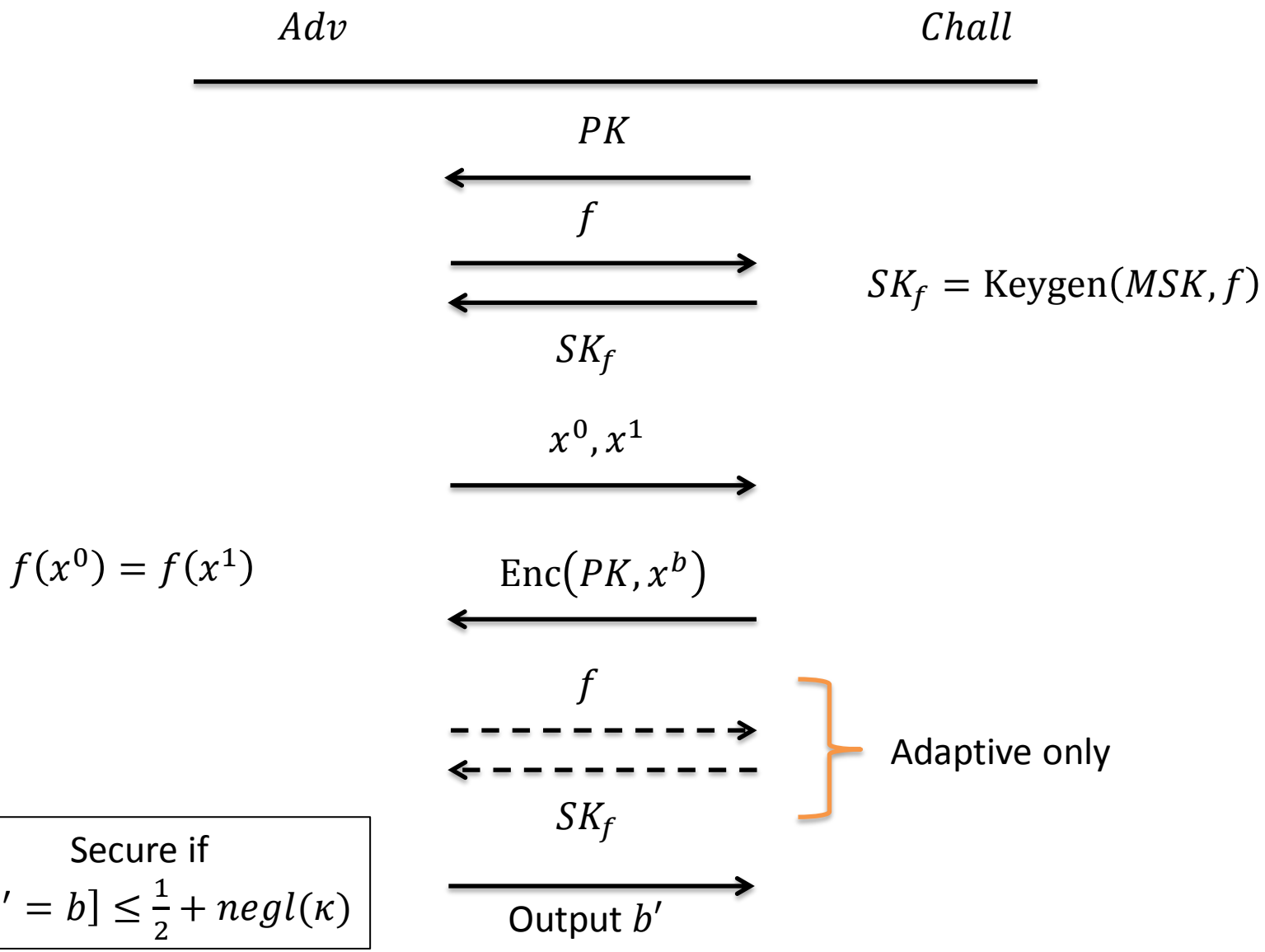
Constructing Func. Cont. HE



Funct. Cont. HE Syntax

Controlled HE	Functional Encryption	Funct. Cont. HE
$(PK, MSK) \leftarrow \text{Setup}(1^n)$	$(PK, MSK) \leftarrow \text{Setup}(1^n)$	$(PK, MSK) \leftarrow \text{Setup}(1^n)$
$c \leftarrow \text{Enc}(PK, x)$	$c \leftarrow \text{Enc}(PK, x)$	$c \leftarrow \text{Enc}(PK, x)$
$EvalK_g \leftarrow \text{EvalKGen}(MSK, g)$		$EvalK_g \leftarrow \text{EvalKGen}(MSK, g)$
	$SK_f \leftarrow \text{Keygen}(MSK, f)$	$SK_f \leftarrow \text{Keygen}(MSK, f)$
$c' \leftarrow \text{HEval}(PK, EK_g, c)$		$c' \leftarrow \text{HEval}(PK, EvalK_g, c)$
$\boldsymbol{g(x)} \leftarrow \text{Dec}(MSK, c')$	$\boldsymbol{f(x)} \leftarrow \text{Dec}(SK_f, c)$	$\boldsymbol{f(g(x))} \leftarrow \text{Dec}(SK_f, c')$

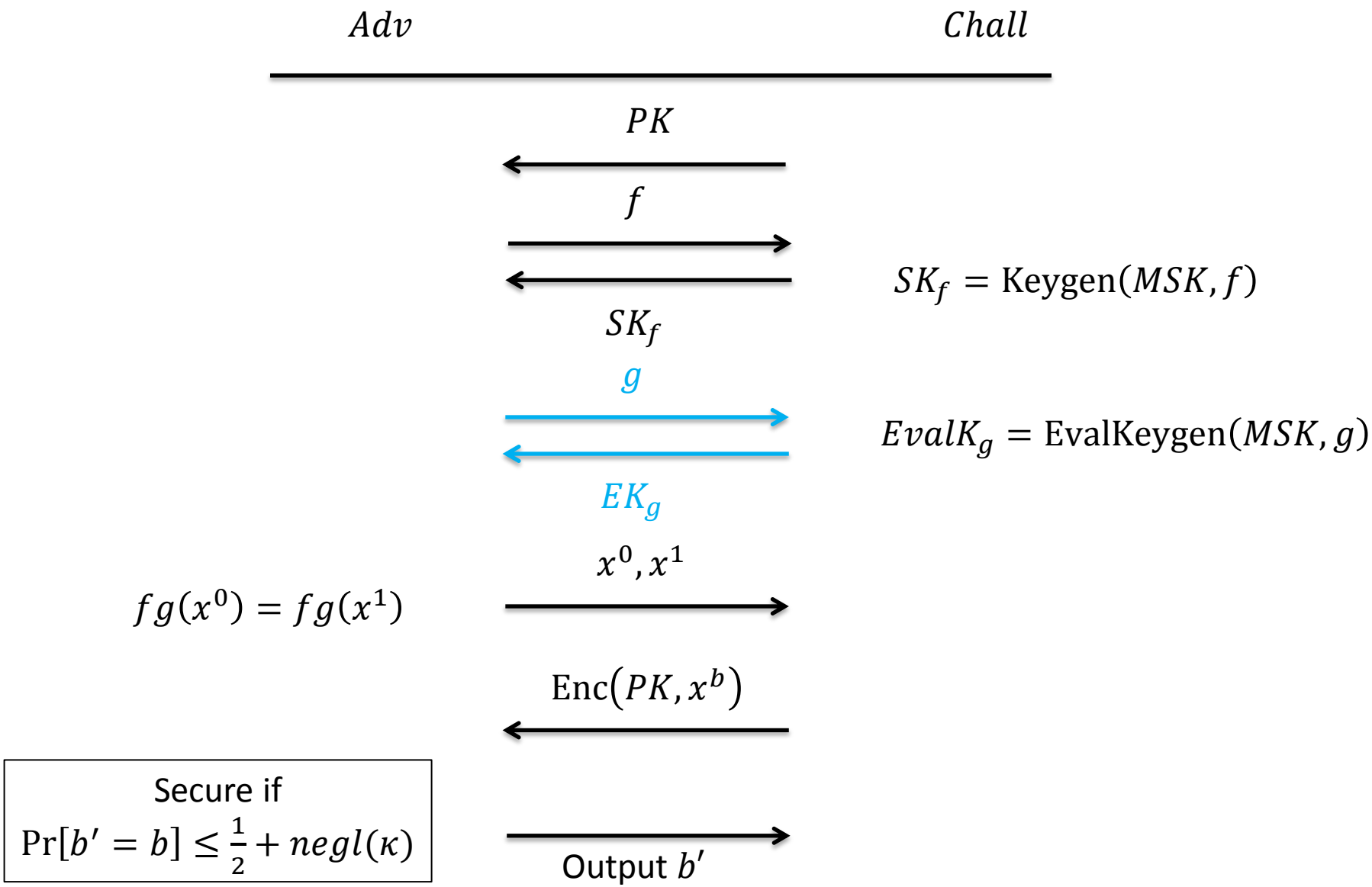
Functional Encryption Security Game



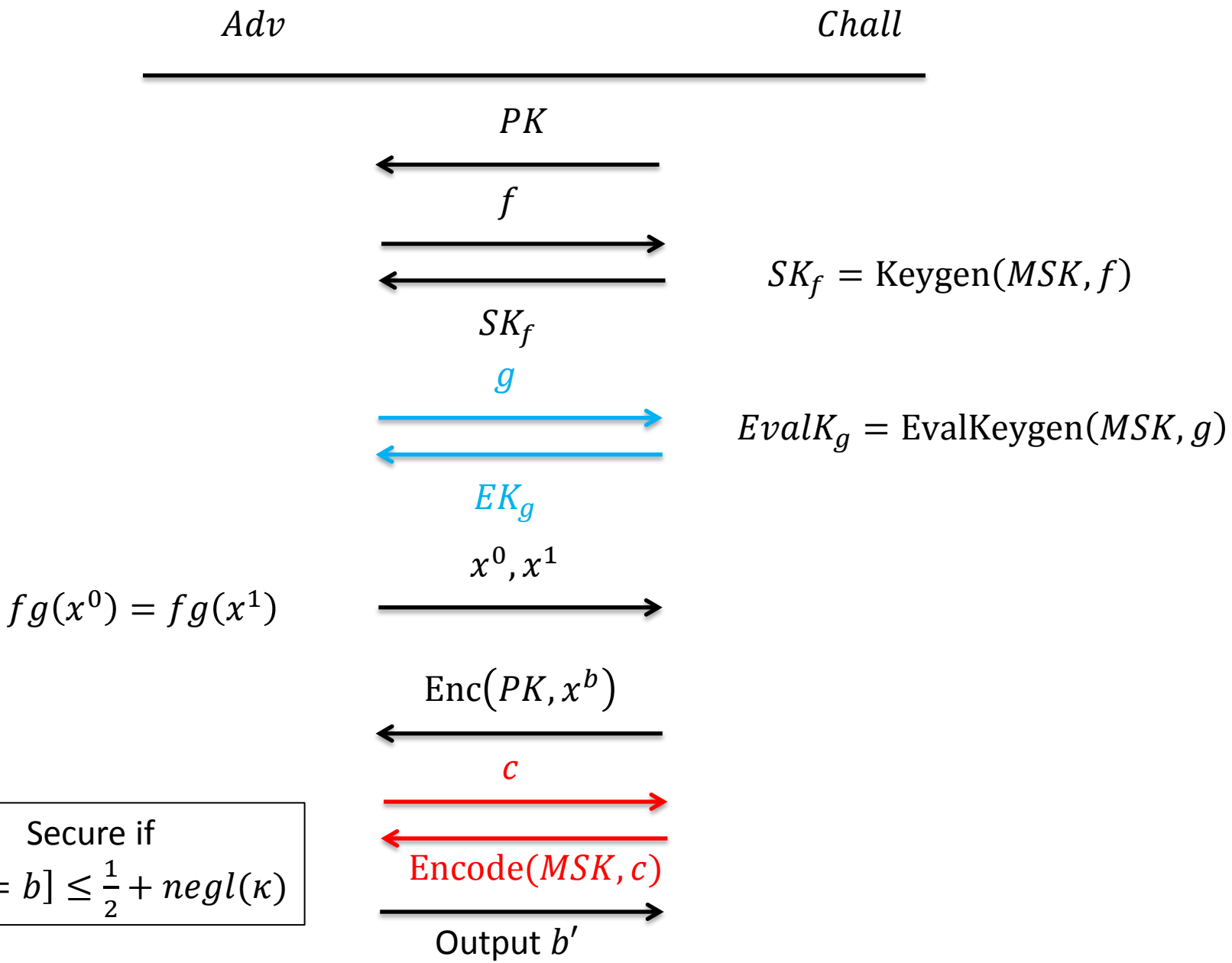
Secure if

$\Pr[b' = b] \leq \frac{1}{2} + negl(\kappa)$

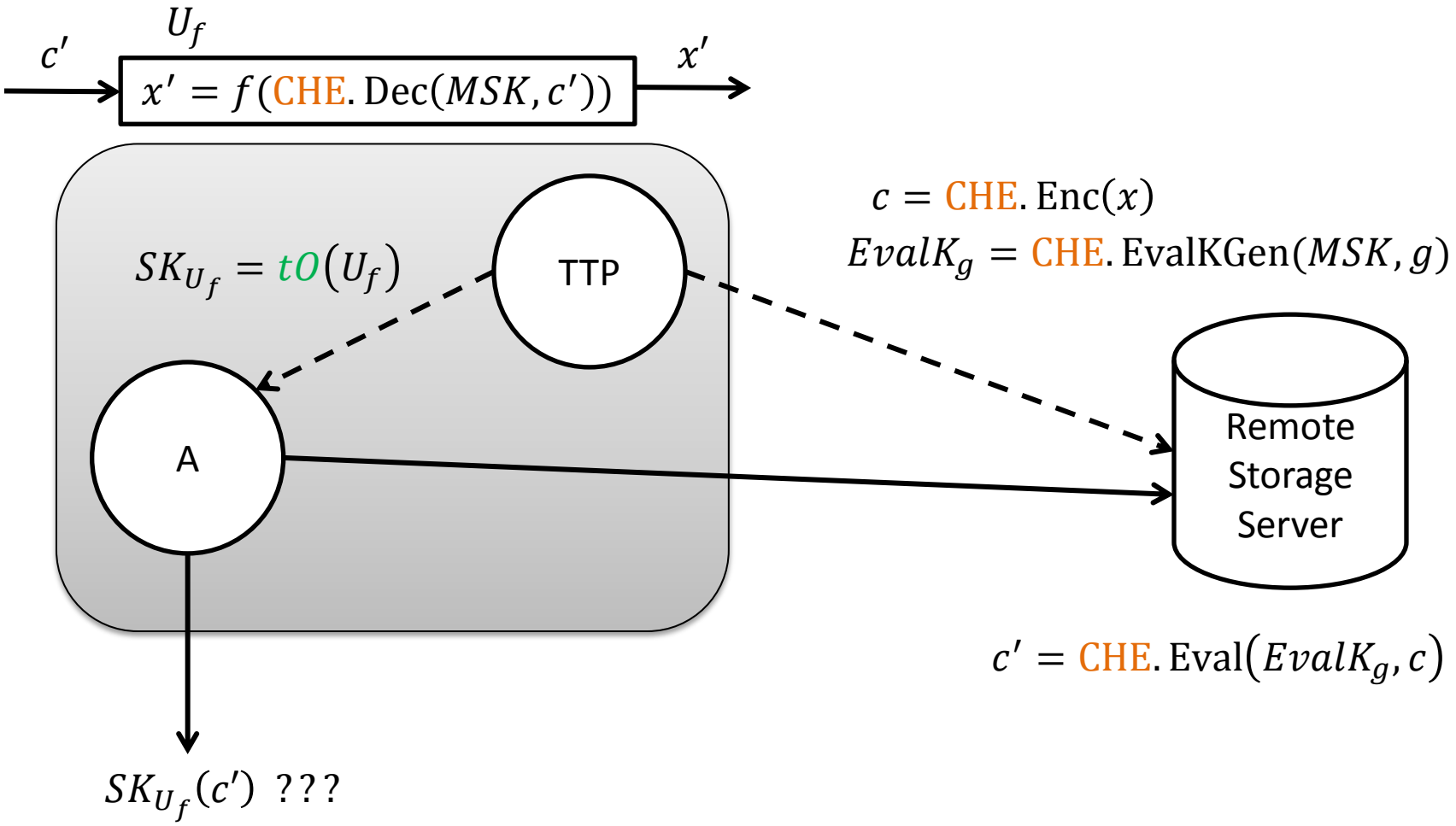
Func. Cont. HE Security Game



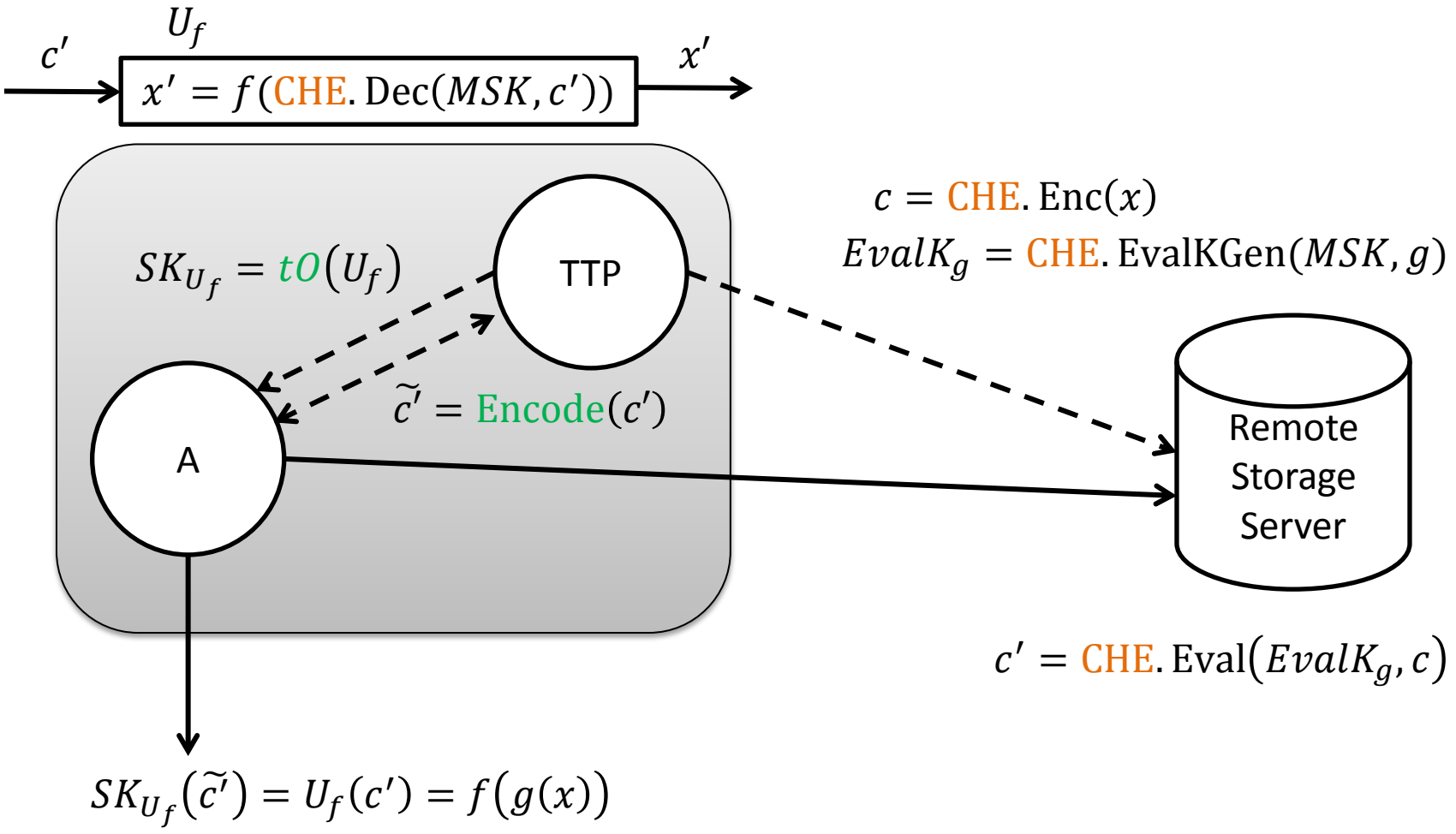
Weak Func. Cont. HE Security Game



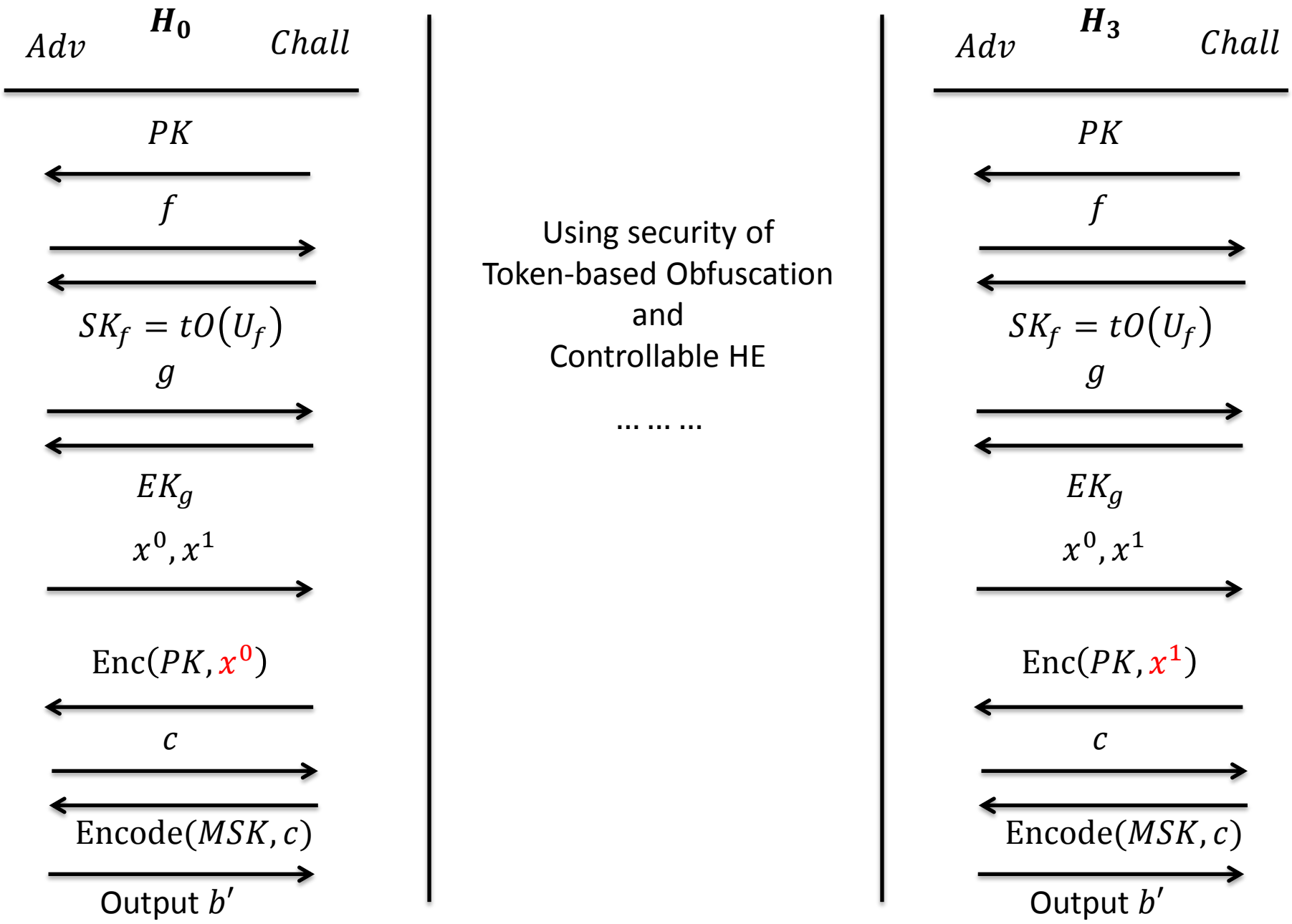
Controllable HE and Token-based Obfuscation → Func. Cont. HE



Controllable HE and Token-based Obfuscation → Func. Cont. HE



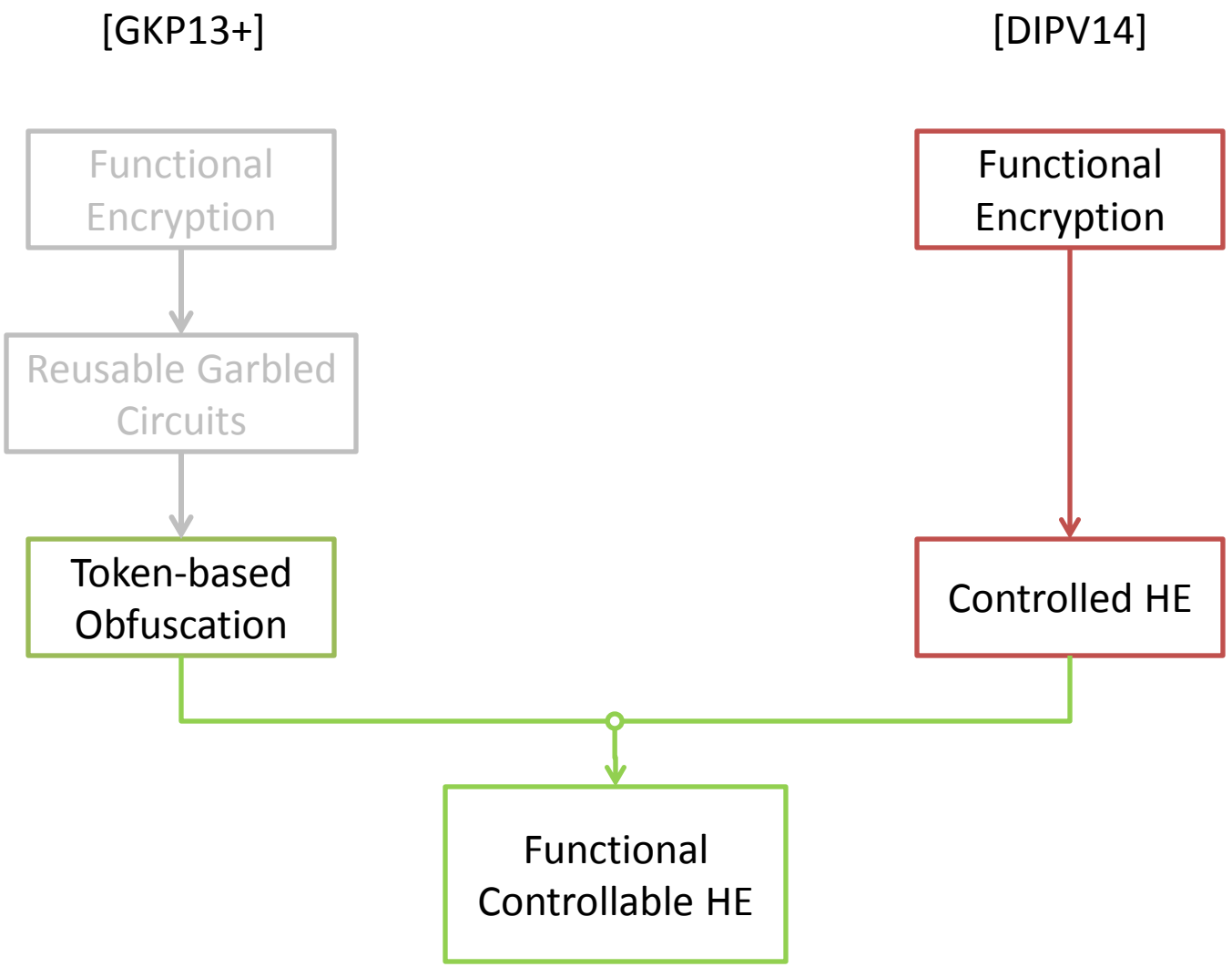
Sketch of Proof



Related Problems to Func. Cont. HE

- Multi-hop Homomorphism
  - Evaluated ciphertexts can be re-evaluated
  - $f\left(g\left(g\left(g(x)\right)\right)\right)$
- Functional Composability
  - Compose multiple functions
  - $f(g_3(g_2(g_1(x))))$
- Multi-input Homomorphism
  - $f\left(g(x_1, \ldots, x_n)\right)$

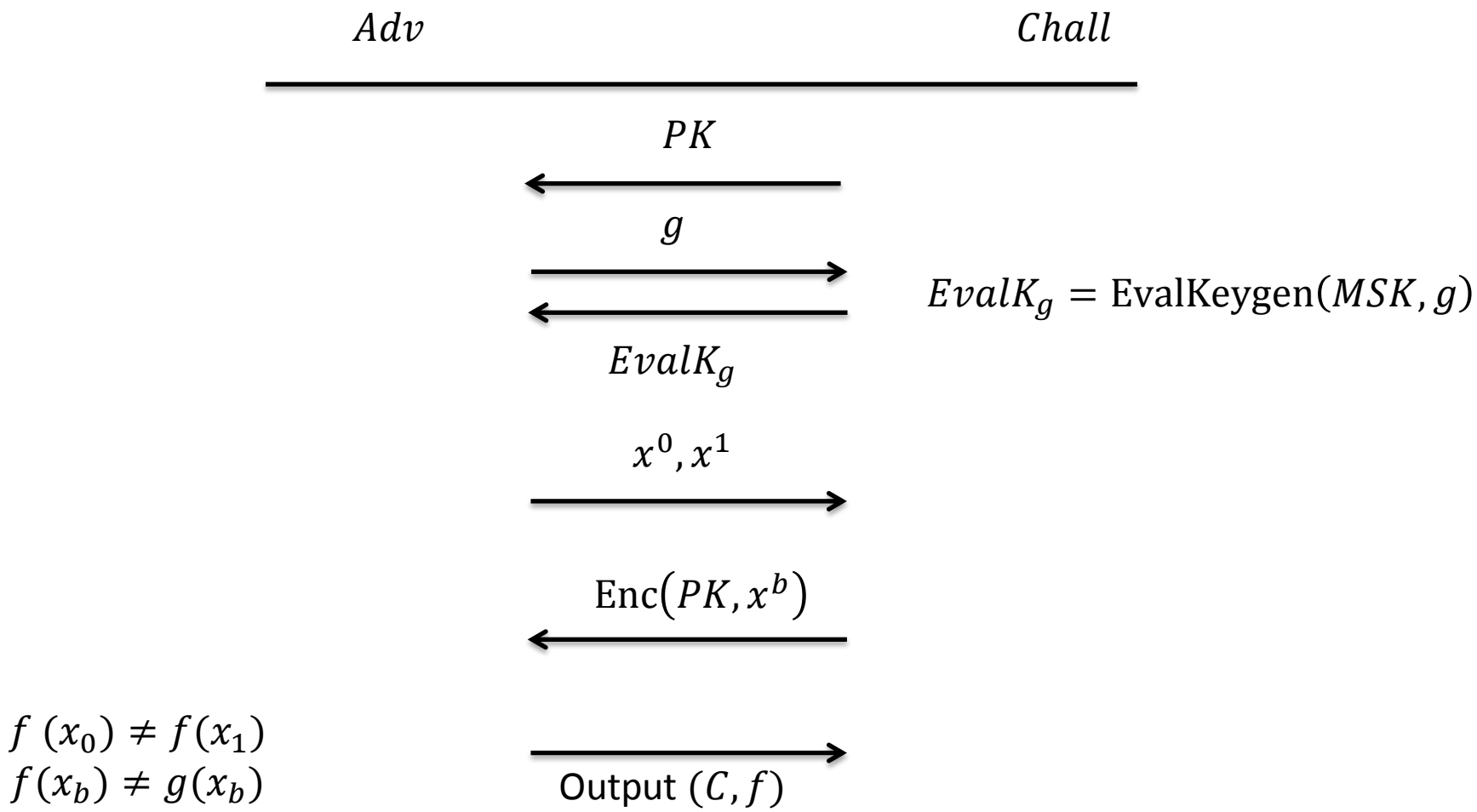
Next: Controlled Homomorphism



Controlled Homomorphic Syntax

Controlled Homomorphic Encryption	**Fully Homomorphic Encryption**
$(MSK, PK) \leftarrow \text{Setup}(1^n)$	$(PK, SK) \leftarrow \text{Setup}(1^n)$
$EvalK_g \leftarrow \text{EvalKeygen}(MSK, g)$	
$c \leftarrow \text{Enc}(PK, x)$	$c \leftarrow \text{Enc}(PK, x)$
$c' \leftarrow \text{HEval}(PK, EK_g, c)$	$c' \leftarrow \text{Eval}(PK, g, c)$
$g(x) \leftarrow \text{Dec}(MSK, c')$	$g(x) \leftarrow \text{Dec}(SK, c')$

Controlled HE Security Game



Secure if

$$\Pr[C = \text{Enc}(f(x_b))] \leq \frac{1}{2} + negl(\kappa)$$

Trivial construction
PKE + Sign → CHE

$CHE.\text{Setup} \rightarrow Pk = ek, \qquad MSK = (dk, sk)$

$CHE.\text{KeyGen}(MSK, C) \rightarrow EvalK_f = \big(f, \text{Sign}(sk, f)\big)$

$CHE.\text{Enc}(Pk, M) \rightarrow C = \text{Enc}_{ek}(M)$

$CHE.\text{HEval}(Ct, EK_c) \rightarrow \big(C, EvalK_f\big) = \Big(C, \big(f, \text{Sign}(sk, f)\big)\Big)$

$CHE.\text{Dec}(dk, Ct) \rightarrow PKE.\text{Dec}_{dk}(C)$

$FE + PRF \rightarrow CHE$

- Two components:
  - Functional Encryption
    $FE = (FE.\text{Setup}, FE.\text{Enc}, FE.\text{KeyGen}, FE.\text{Eval})$
  - Pseudorandom Function
    $\mathcal{F} = \{F(.,.)\}$

High-level Idea

- $\mathrm{CHE.Setup}(1^n) \rightarrow (Pk, MSK)$
- $\mathrm{CHE.Enc}(Pk, M) \rightarrow C$
- $\mathrm{CHE.KeyGen}(\mathrm{MSK}, \mathrm{f}) \rightarrow \mathrm{EvalK_f}$
- $\mathrm{CHE.HEval}\left(Pk, f, EK_f\right) \rightarrow C'$
- $\mathrm{CHE.Dec}(MSK, C') \rightarrow f(M)$

FE.Setup

FE.Enc

FE.Keygen

FE.Dec

High-level Idea

- $\mathrm{CHE.Setup}(1^n) \rightarrow (Pk, MSK)$
- $\mathrm{CHE.Enc}(Pk, M) \rightarrow C$
- $\mathbf{CHE.KeyGen}(\boldsymbol{MSK}, \boldsymbol{f}) \rightarrow \boldsymbol{EvalK_f}$
- $\mathrm{CHE.HEval}\left(Pk, f, EK_f\right) \rightarrow C'$
- $\mathrm{CHE.Dec}(MSK, C') \rightarrow f(M)$

$f'(M) = \mathrm{FE.Enc}_{pk}(f(M))$

$C' = \mathrm{FE.Enc}(f(M))$

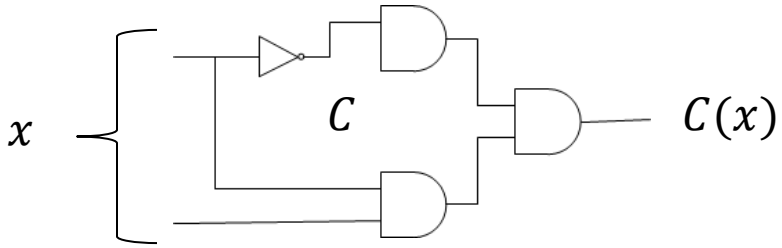
Actual Construction

- $\mathrm{CHE.Setup}\begin{cases} Pk = FE.PK \\ MSK = FE.MSK \end{cases}$
- $\mathrm{CHE.Enc}(M) = \mathrm{FE.Enc}(M, r)$
- $\mathrm{CHE.KeyGen}(MSK, f)\begin{cases} f'(M, r) = \mathrm{FE.Enc}\left(FE.Pk, \left(\boldsymbol{C}(\boldsymbol{M})\right); \boldsymbol{F}(\boldsymbol{r}, \boldsymbol{f})\right) \\ EvalK_f = \mathrm{FE.KeyGen}(f') \end{cases}$
- $\mathrm{CHE.HEval}\left(C, EvalK_f\right) = \mathrm{FE.Dec}\left(C, EvalK_f\right)$
- $\mathrm{CHE.Dec}(MSK, C) = \mathrm{FE.Dec}(C, EvalK_{Id})$

Thank you!

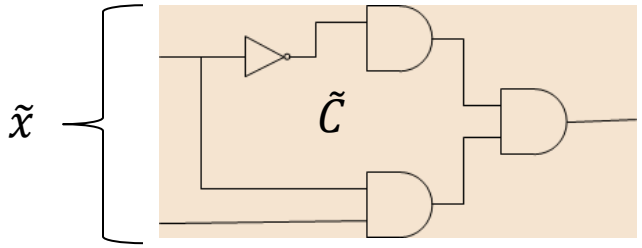
Questions?
Suggestions?

Garbling Mechanisms



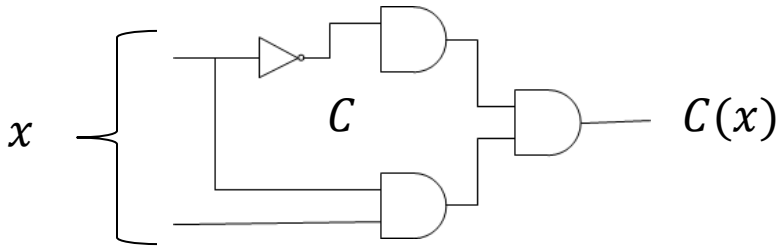
Sender:
$(\tilde{C}, sk) \leftarrow \boldsymbol{GC}(C)$
$\tilde{x} \leftarrow \boldsymbol{GI}(sk, x)$

$(\tilde{C}, \tilde{x})$

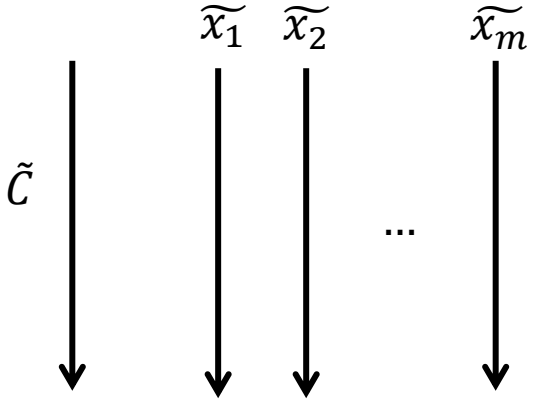


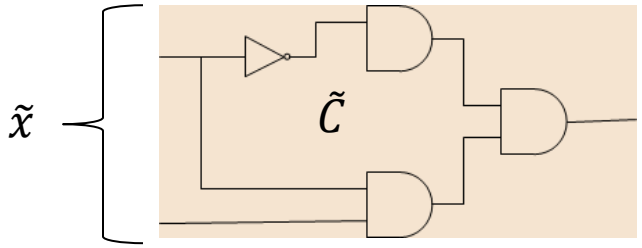
Evaluator:
$\tilde{C}(\tilde{x}) \leftarrow \boldsymbol{GE}(\tilde{C}, \tilde{x})$

Reusable Garbled Circuit



Sender:
$(\tilde{C}, sk) \leftarrow \boldsymbol{GC}(C)$
$\widetilde{x_i} \leftarrow \boldsymbol{GI}(sk, x_i)$





Evaluator:
$\tilde{C}(\tilde{x}) \leftarrow \boldsymbol{GE}(\tilde{C}, \tilde{x})$

FE → Reusable Garbled Circuits

$U_E(k, x)$

$C \leftarrow \text{Sym. Dec}(k, E)$
Output $C(x)$

- $\boldsymbol{GC}(C)$:
  - $(MSK, PK) \leftarrow \text{FE. Setup}$
  - $k \leftarrow \text{Sym. Setup}$
  - $\tilde{C} \leftarrow \text{FE. Keygen}(MSK, U_E)$
  - Output $(\tilde{C}, k)$
- $\boldsymbol{GI}(k, x)$: $\tilde{x} \leftarrow \text{FE. Enc}\big(PK, (k, x)\big)$
- $\boldsymbol{GE}(\tilde{C}, \tilde{x})$: $\text{FE. Dec}\big(\tilde{C}, \tilde{x}\big) = \tilde{C}(k, x) = U_E(k, x) = C(x)$

Sketch of Proof