Efficient Threshold Encryption from Lossy Trapdoor Functions

Xiang Xie, Rui Xue and Rui Zhang

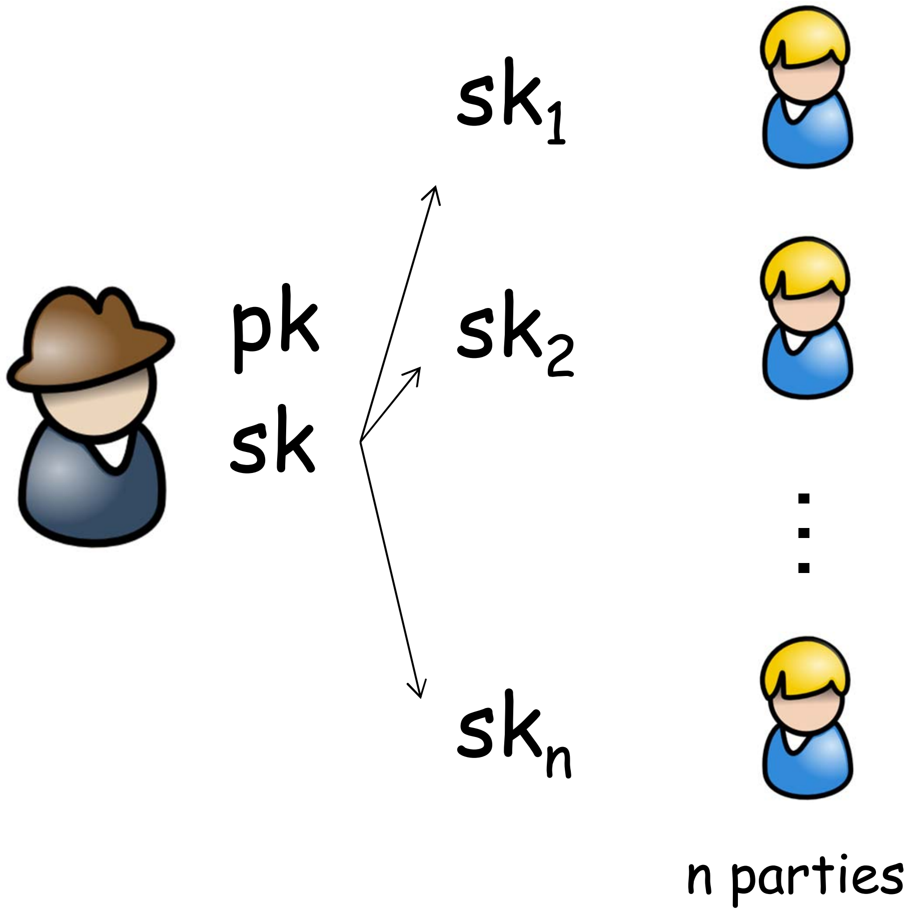
SKLOIS

Chinese Academy of Sciences

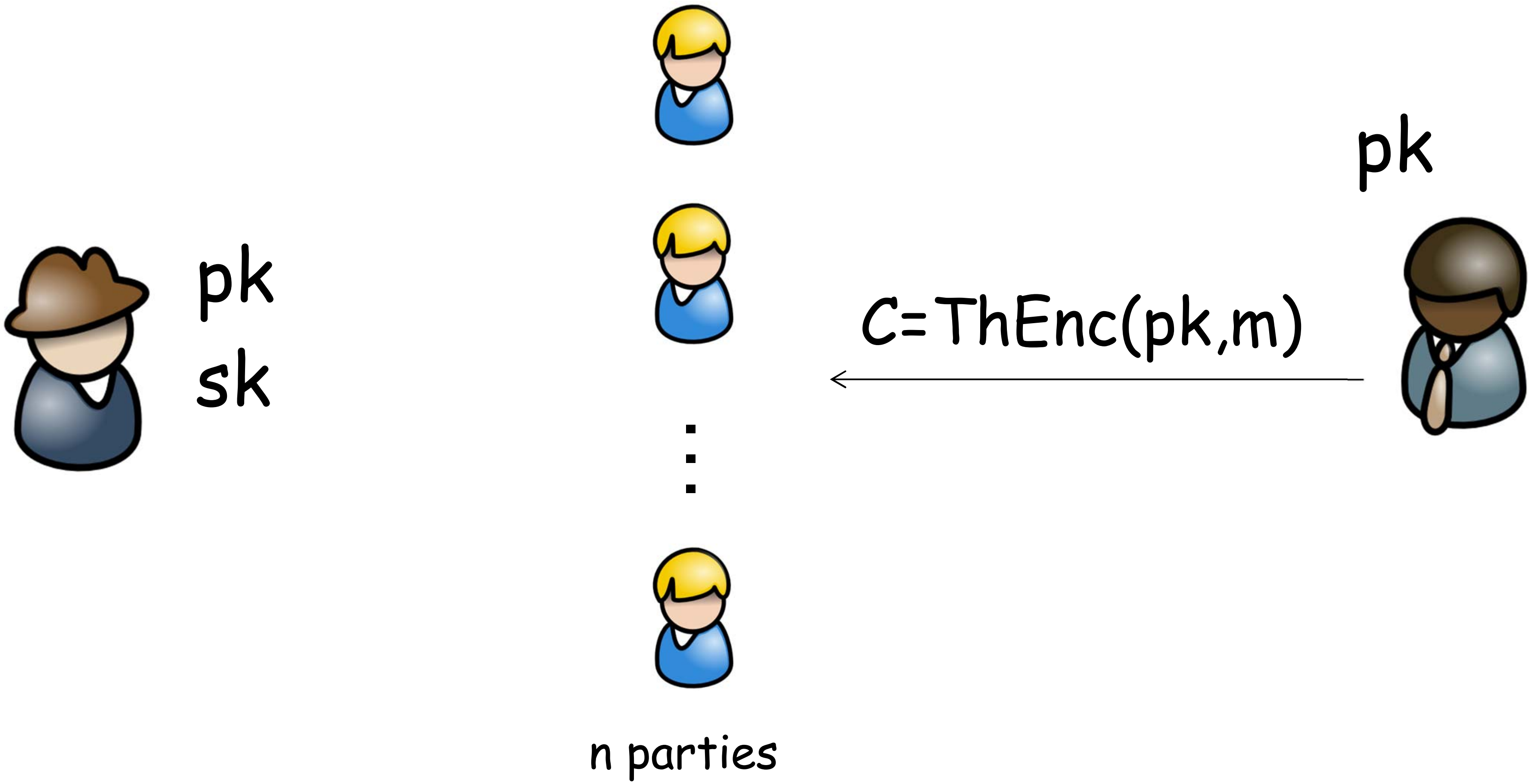
Outline

- Background
- Our Results
- Our Constructions
- Conclusions

Threshold Public Key Encryption (ThPKE)



Threshold Public Key Encryption (ThPKE)

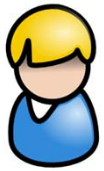


Threshold Public Key Encryption (ThPKE)

If more than t_p parties are honest
$m = \mathrm{Combine}(m_1, m_2, \ldots, m_n)$

pk
sk

$m_1 = \mathrm{ThDec}(C, sk_1)$



$m_2 = \mathrm{ThDec}(C, sk_2)$

⋮

$m_n = \mathrm{ThDec}(C, sk_n)$

n parties

pk

Formal definition

ThPKE=(ThGen, ThEnc, ThDec ThCom)

- ThGen: $(pk, \overrightarrow{sk}) \longleftarrow$ ThGen(λ, n, t_p)
- ThEnc: $C \longleftarrow$ ThEnc(pk,m)
- ThDec: $m_i \longleftarrow$ ThDec(sk_i, C)
- ThCom: $m \longleftarrow$ ThCom($m_1, m_2, \ldots, m_n$)

Security

Announce threshold t_p to be corrupted

pk sk_1, sk_2 ,…, sk_{tp}

(i , C)

Static Attacker

m_i=ThDec(C, sk_i)

Challenger

⋮

m_0, m_1

C^*=ThEnc(pk, m_b), $b \leftarrow \{0,1\}$

(i , $C \neq C^*$)

m_i=ThDec(C, sk_i)

⋮

Output b' (guess b)

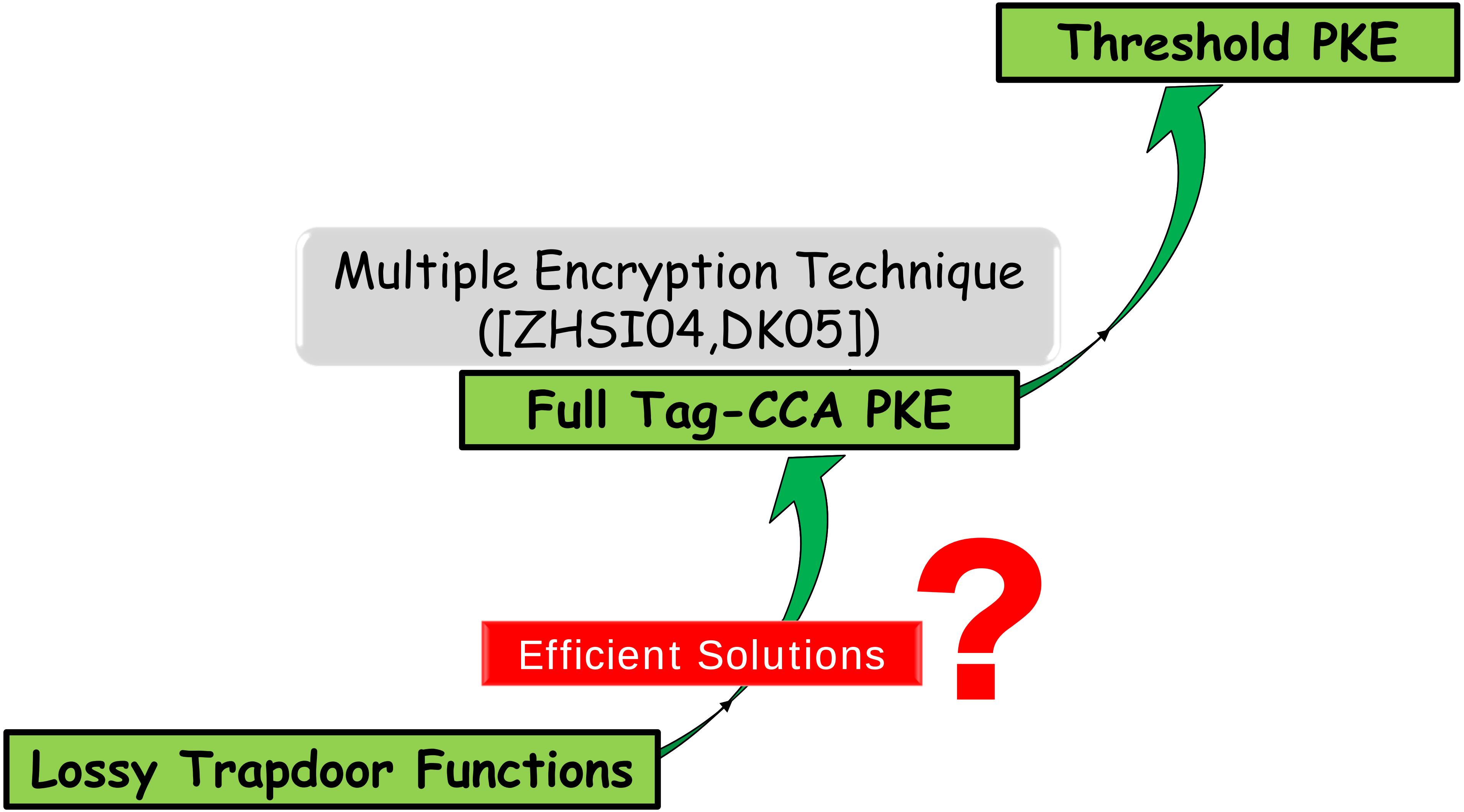
Related work

- Introduced by Desmedt'87 and Desmedt-Frankel'90
- Shoup-Gennaro'98 (ROM)
- Canetti-Goldwasser'99 (interactive or storage of secrets)
- Zhang-Hanaoka-Shikata-Imai'04,Dodis-Katz'05 (generic constructions from ME)
- Boneh-Boyen-Halevi'05, Arita–Tsurudome'09 (pairing)
- Bendlin-Damgard'10 (lattice, not generic)

Overview of our results

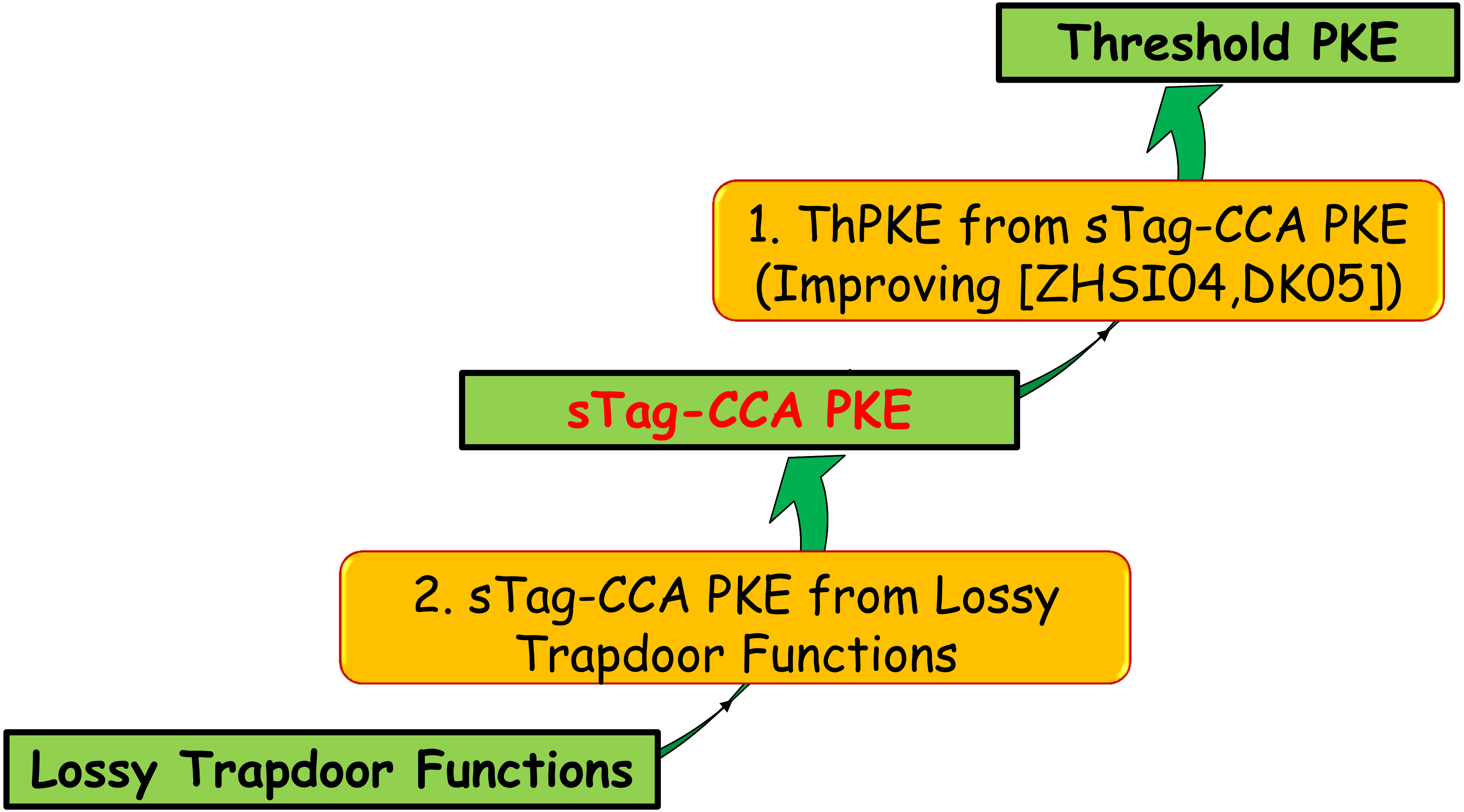
1. Generic threshold public encryption
   - Inspired from Dodis-Katz'05
   - Weaker components than those in DK'05
     - sTag-CCA instead of Tag-CCA
2. sTag-CCA PKE from lossy trapdoor functions
   - ThPKE from lattices (against quantum attackers)
3. Comparisons with other schemes from Lattice
   - slightly efficient than the known lattice based scheme (BD'10)

Basic Ideas



Threshold PKE
Multiple Encryption Technique ([ZHSI04,DK05])
Full Tag-CCA PKE
Efficient Solutions
?
Lossy Trapdoor Functions

Towards our goal...



Ingredients

- Tag-based PKE (TPKE)

Informally, the encryption and the decryption algorithms take an additional input: a "tag" (denoted as τ).

- TPKE=(TGen, TEnc, TDec)
  - $(pk,sk) \leftarrow TGen(k)$
  - $(C, \tau) \leftarrow TEnc(pk, \tau, m)$
  - $m \leftarrow TDec(sk, C, \tau)$

Security of TPKE

- Full Tag-CCA (used in DK'05)
  - $(C, \tau) \neq (C^*, \tau^*)$ in 2nd CCA-query stage
  - (C, τ^*) is a legal query as long as $C \neq C^*$
- sTag-CCA
  - $\tau \neq \tau^*$ for a query (C, τ) in 2nd CCA-query stage
  - Any (C^*, τ) with $\tau \neq \tau^*$ is a legal query

sTag-CCA is a weaker security defnition than full Tag-CCA !

Other ingredients

- Secret Share scheme SS = (Share, Rec) with privacy threshold t_p
  - $(m_1, m_2, \ldots, m_n) \leftarrow Share(m, n)$
  - $m \leftarrow Rec(m_1, m_2, \ldots, m_n)$
  - t_p legal shares do not reveal any information of m
- Signature scheme Σ=(Gen, Sign, Ver)
  - Strongly unforgeable one-time signature
    - An attacker is able to make at most one query to the sign oracle on a message m, and obtain σ.
    - The attacker wins if he outputs $(m^*, \sigma^*) \neq (m, \sigma)$ and $Ver(m^*, \sigma^*) = 1$

Construction: step 1

$$\text{“}\underbrace{SS + TPKE + Sig}_{\text{Step 1}} = ThPKE\text{”}$$

Security of TPKE

Select τ^* to the challenger

pk

$(C, \tau \neq \tau^*)$

Selective Attacker

$m = TDec(sk, C, \tau)$

Challenger

⋮

m_0, m_1

$(C^*, \tau^*) = TEnc(pk, \tau^* \; m_b)$ $\quad b \leftarrow \{0,1\}$

$(C, \tau \neq \tau^*)$

$m = TDec(sk, C, \tau)$

⋮

Output b' (guess b)

Intuition of the design of DK'05

c_1 = TEnc(pk_1, svk, m_1)

c_2 = TEnc(pk_2, svk, m_2)

⋮

c_n = TEnc(pk_n, svk, m_n)

σ = Sign(ssk, $(c_1,\ldots c_n)$)

$c = \langle svk, c_1, c_2, \ldots, c_n, \sigma \rangle$

The adversary can no longer modify the ciphertext!

Our construction

- Given TPKE=(TGen, TEnc, TDec), SS = (Share, Rec) Σ = (Gen, Sign, Ver), we construct

ThPKE=(ThGen,ThEnc, ThDec, ThCom) as follows.

- ThGen(n, t_p)
  - $(pk_1,sk_1) \leftarrow$ TGen, …, $(pk_n,sk_n) \leftarrow$ TGen,
  - Set $PK=(pk_1,\ldots, pk_n)$, $Sk_i=sk_i$

- ThEnc(PK, m)
  - $(m_1,\ldots,m_n)$=Share(m); $(svk,ssk) \leftarrow$ Gen
  - c_1 = TEnc(pk_1, svk, m_1),…, c_n = TEnc(pk_n, svk, m_n)
  - σ = Sign(ssk, $(c_1,\ldots c_n)$)
  - Output $C=(svk, c_1,\ldots c_n, \sigma)$

Our construction

- ThDec(Sk_i, C)
  - Parse C = (svk, c_1,…c_n, σ)
  - Check Ver(svk, (c_1,…c_n)) =1; if not, abort
  - Output m_i = TDec(sk_i, c_i ,svk)

- ThCom(m_1,…,m_n)
  - Output m=Rec(m_1,…,m_n)

Security of our scheme

Theorem 1. ThPKE constructed above is a CCA secure threshold encryption scheme, if TPKE is sTag-CCA secure, SS is t_p secure and $\sum$ is one-time strongly unforgeable.

Proof sketch: We define a sequence of games to prove this theorem.

W.l.o.g we assume $\{n-t_p+1,\ldots n\}$ are corrupted.

1, If decryption query C is of the form (svk^*, $c_1, \ldots c_n$ σ), abort.
This can be done via the one-time strongly unforgeable signature.

Security of our scheme

2. For $1 \leq i \leq n - t_p - 1$, the challenger change the challenge ciphertext as:

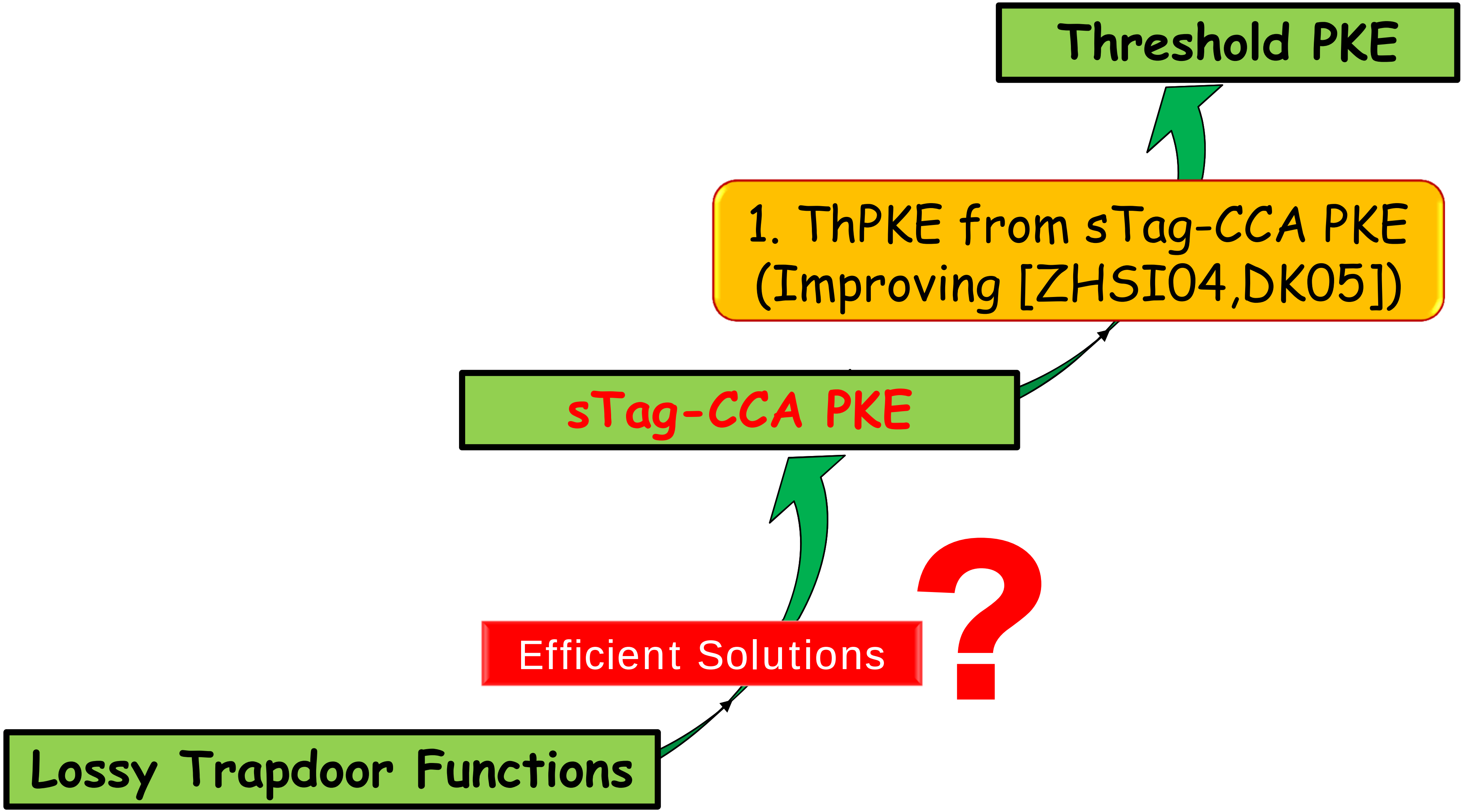
Game i: $(TEnc(pk_1,0), \ldots, TEnc(pk_i, 0), \boxed{TEnc(pk_{i+1},m_{i+1})}, \ldots, TEnc(pk_n,m_n))$

Game i+1: $(TEnc(pk_1,0), \ldots, TEnc(pk_i, 0), \boxed{TEnc(pk_{i+1},0)}, \ldots, TEnc(pk_n,m_n))$

View(Game i) $\approx$ View(Game i+1)

according to the sTag-CCA of TPKE scheme !

Up to now...


Threshold PKE
1. ThPKE from sTag-CCA PKE
(Improving [ZHSI04,DK05])
sTag-CCA PKE
Efficient Solutions
?
Lossy Trapdoor Functions

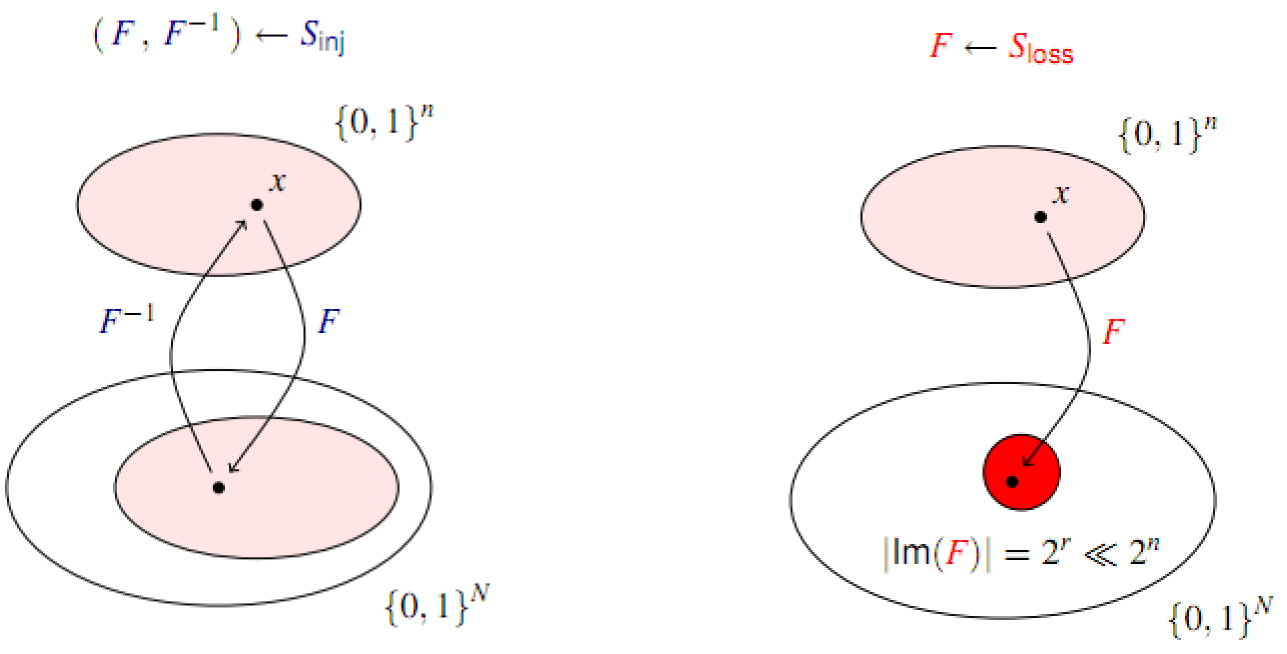
Construction: step 2

How to sTag-CCA PKE

We obtain sTag-CCA PKE from lossy trapdoor functions and All-But-One (ABO) trapdoor functions [PK'08].

Lossy trapdoor functions

$$F \overset{c}{\approx} F$$



All-But-One trapdoor functions

"LF + Additional Branch Set"

$(s,td) \leftarrow S_{abo}(b^*)$

$G(s,b,x)$: an injective trapdoor function (with $b \neq b^*$)

$G(s,b^*,x)$: a lossy function

$$s_0 \approx s_1$$

$$(s_0,td_0) \longleftarrow S_{abo}(b_0),\ (s_1,td_1) \longleftarrow S_{abo}(b_1)$$

For any b_0,b_1

Our sTag-CCA PKE

PKE = (Gen, Enc, Dec)

- Gen(k)
  - $(F, F^{-1}) \leftarrow S(inj,k)$, $(s, td) \leftarrow S_{abo}(0,k)$,
  - Sample a pairwise independent hash h
  - pk=(F,G, h), sk=(F^{-1}) (td' for proof)

- Enc (m)
  - Choose b (tag) from the branch set.
  - Randomly choose x (compactible with F and G)
  - C=< F(x), G(s, b, x), h(x) XOR m >
  - Output (C, b)

Our sTag-CCA PKE

- Dec (C, b)
  - Parse C as (c_1, c_2, c_3)
  - $x = F^{-1}(c_1)$
  - Check $F(x) = c_1$, $G(s, x, b) = c_2$; If not, abort
  - Output x XOR c_3

It is exactly the Peikert-Waters "basic PKE" from LTFs !

In [PW08], it was proved that this construction is CCA1 secure.

Our sTag-CCA PKE

Theorem 2. The encryption scheme PKE=(Gen, Enc, Dec) described above is sTag-CCA secure.

Proof sketch

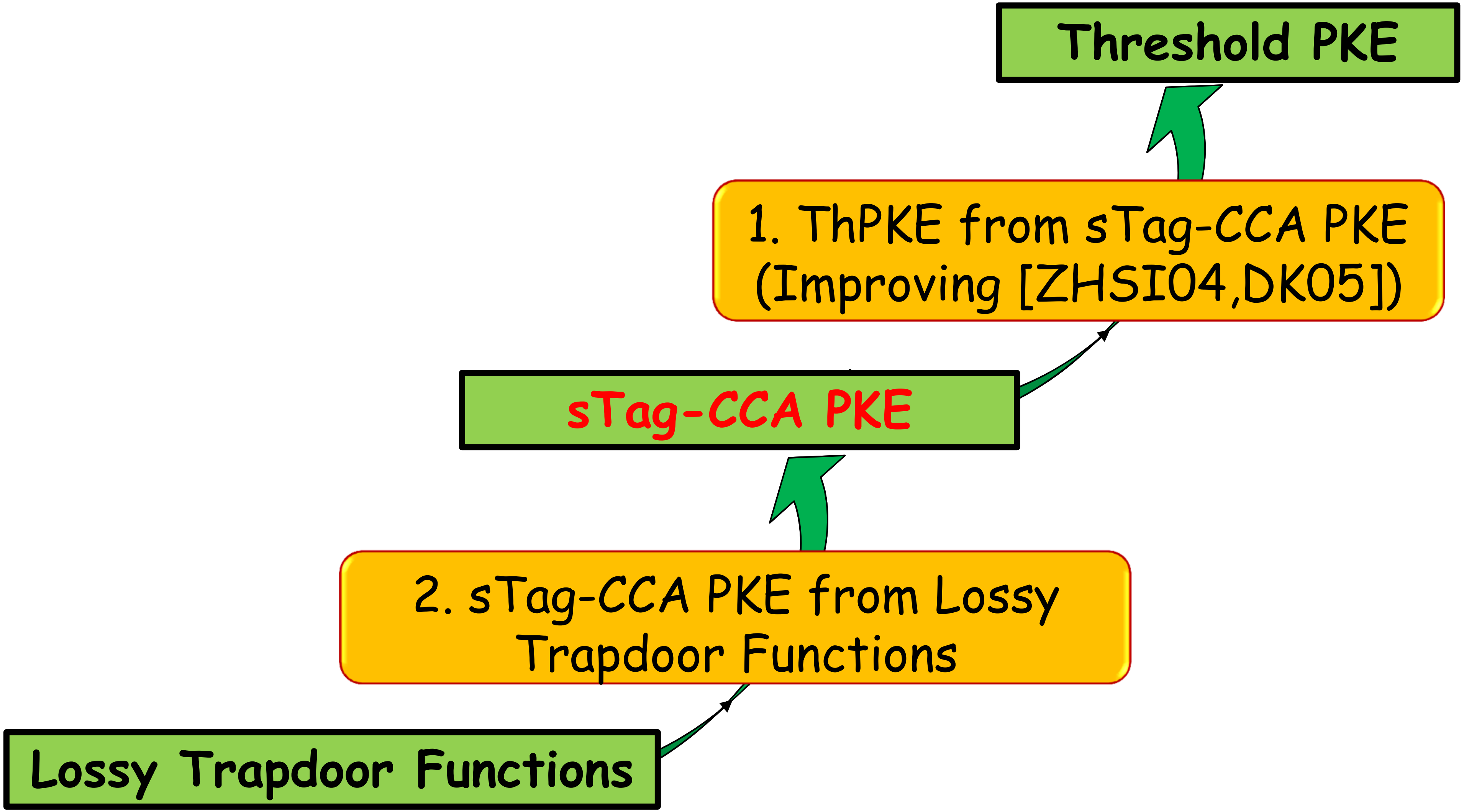
Game 1: $(s, td) \longleftarrow S_{abo}(b^*)$ instead of $(s, td) \longleftarrow S_{abo}(0)$

Game 2: use td to answer decryption queries.

Game 3: $(s, *) \longleftarrow S(lossy)$ instead of $(s, td) \longleftarrow S(inj)$

Game 4: use randomly chosen r instead of c_3^*

Wrapping up the whole story...



Comparisons of ThPKE

Table 1. Comparisons among schemes

Schemes	PK Size	SK Size of Each Sever	Ciphertext Size	Assumption	RO Free	Quantum Attack Resistance
SG98	$(n+2)\|\mathbb{G}\|$	$\|\mathbb{Z}_q\|$	$5\|\mathbb{G}\| + 2\|\mathbb{Z}_q\|$	CDH	$\times$	$\times$
CG99	$5\|\mathbb{G}\|$	$(L+5)\|\mathbb{Z}_q\|$	$4\|\mathbb{G}\|$	DDH	$\surd$	$\times$
BBH06	$(n+4)\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$2\|\mathbb{G}\| + \|\mathbb{G}_T\| + \|\mathtt{SIGN}\|$	DBDH	$\surd$	$\times$
AT09	$(n+4)\|\mathbb{G}\|$	$\|\mathbb{Z}_q\|$	$2\|\mathbb{G}\| + \|\mathbb{G}_T\| + \|\mathtt{SIGN}\|$	DBDH	$\surd$	$\times$
BD10	γ^5	$(2n-1)\gamma$	γ^2	SIVP_{γ^4}	$\surd$	$\surd$
Ours	$2n\gamma^3 \log \gamma$	$\gamma^2 \log \gamma$	$2n\gamma^2 \log \gamma$	$\mathrm{SIVP}_{\delta\gamma}$	$\surd$	$\surd$

Conclusions

- ThPKE from LTFs
  1. ThPKE from sTag-CCA PKE
  2. sTag-CCA PKE from LTFs
- Concrete implementation from Lattices
  - (Slightly) better than the previous one from lattice [BD'10]