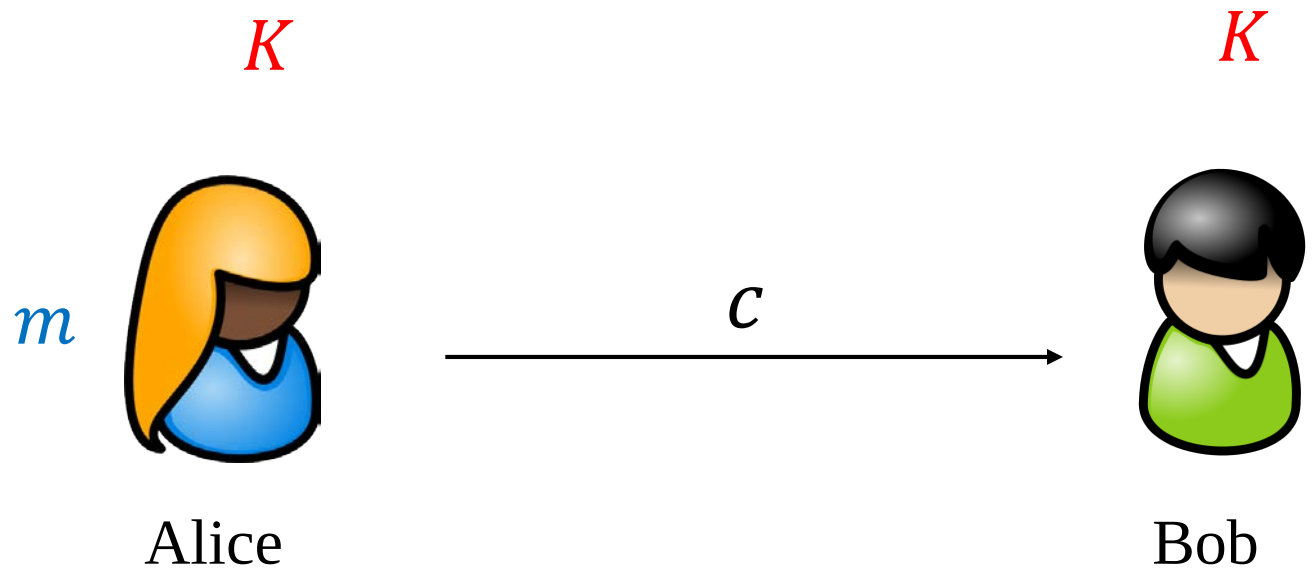
Identity Based Encryption from the Diffie-Hellman Assumption

Sanjam Garg

University of California, Berkeley

(Joint work with Nico Döttling)

Private-Key Encryption



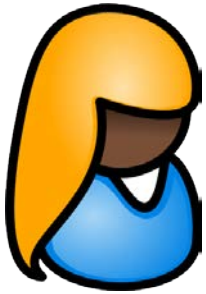
$$c = Enc(K, m)$$

Public-Key Encryption [DH76,RSA78,GM82]

Obtain pk_{Bob}

sk_{Bob}

m



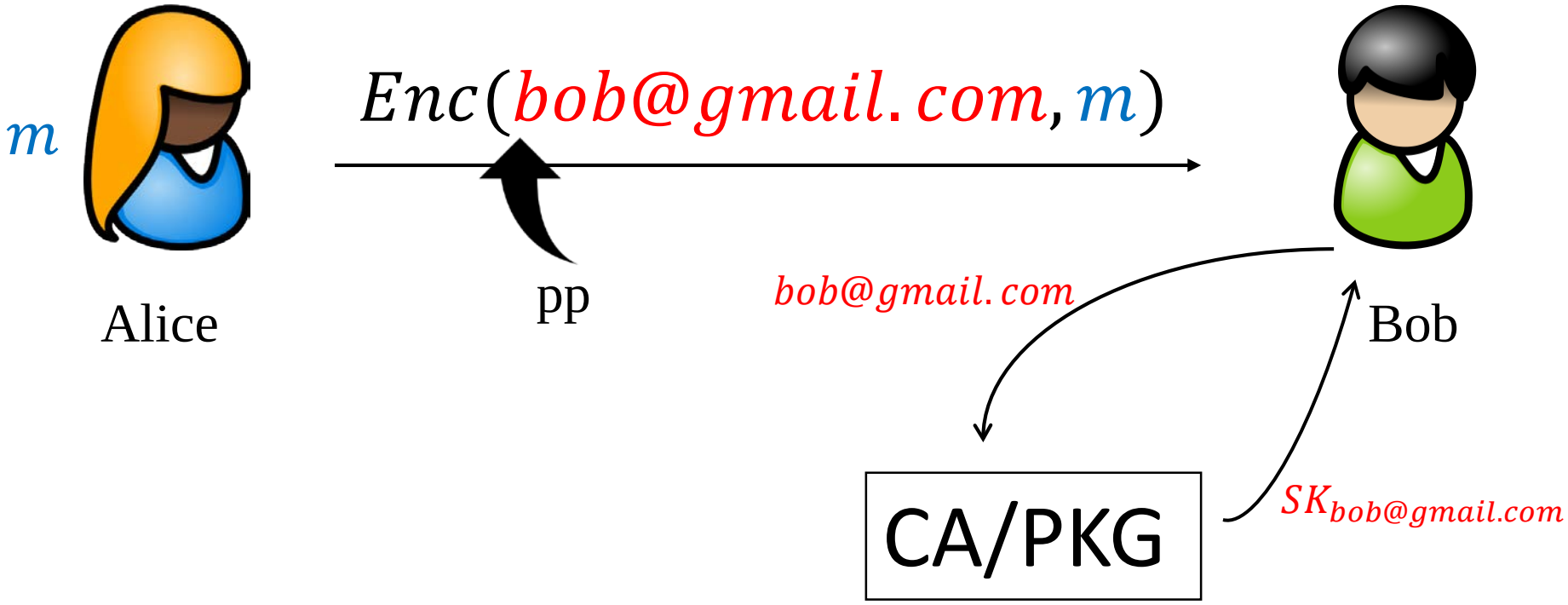
$Enc(pk_{Bob}, m)$

Alice

Bob

Identity-Based Encryption (IBE) [Shamir84]

Identity of the recipient used as the public key



Identity-Based Encryption (IBE) [Shamir84]

Four Algorithms: (S, K, E, D)

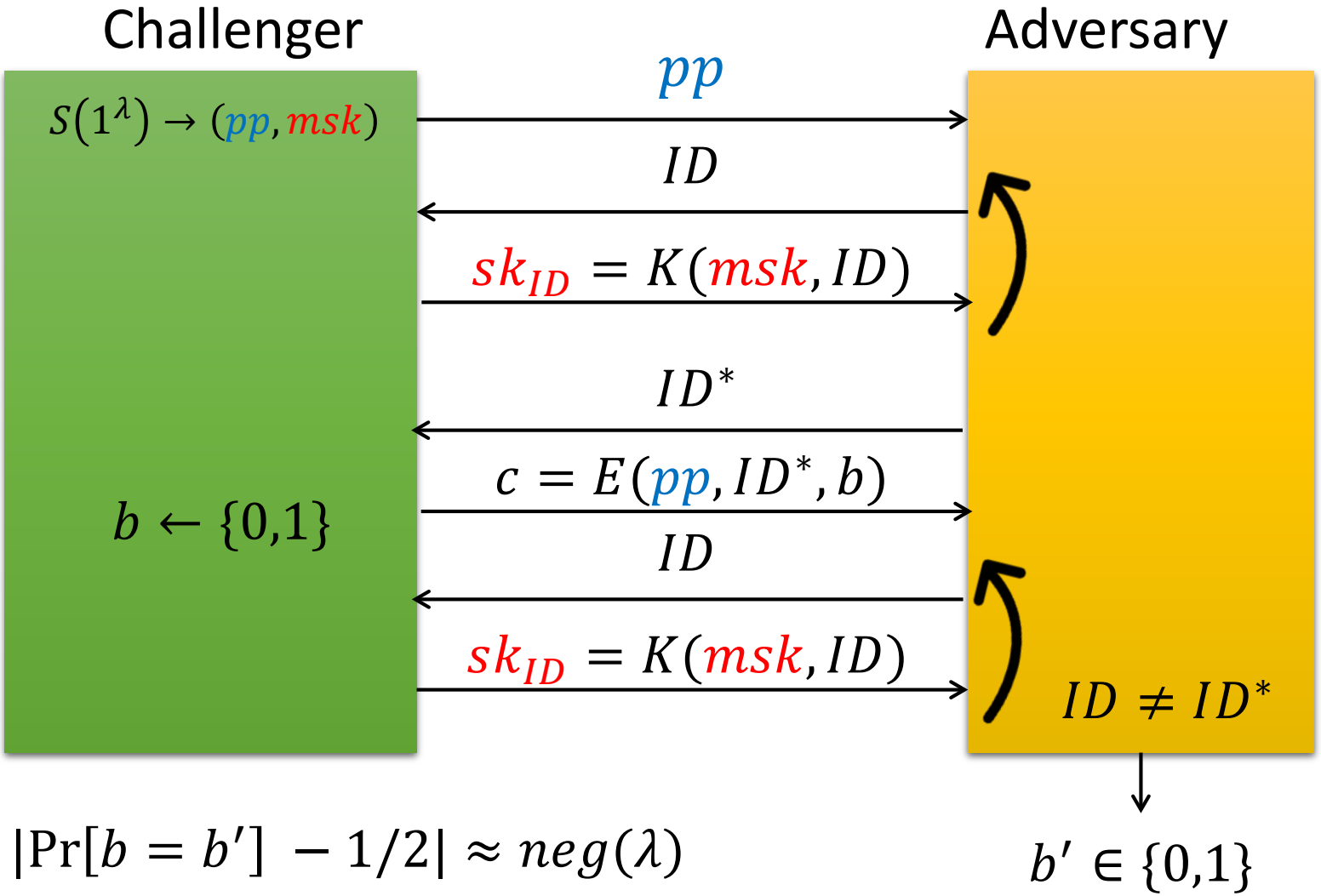
$S(1^\lambda) \rightarrow (pp, msk)$ pp are public parameters
msk is the master secret-key

$K(msk, ID) \rightarrow sk_{ID}$ sk_{ID} secret key for ID

$E(pp, ID, m) \rightarrow c$ encrypt using pp and ID

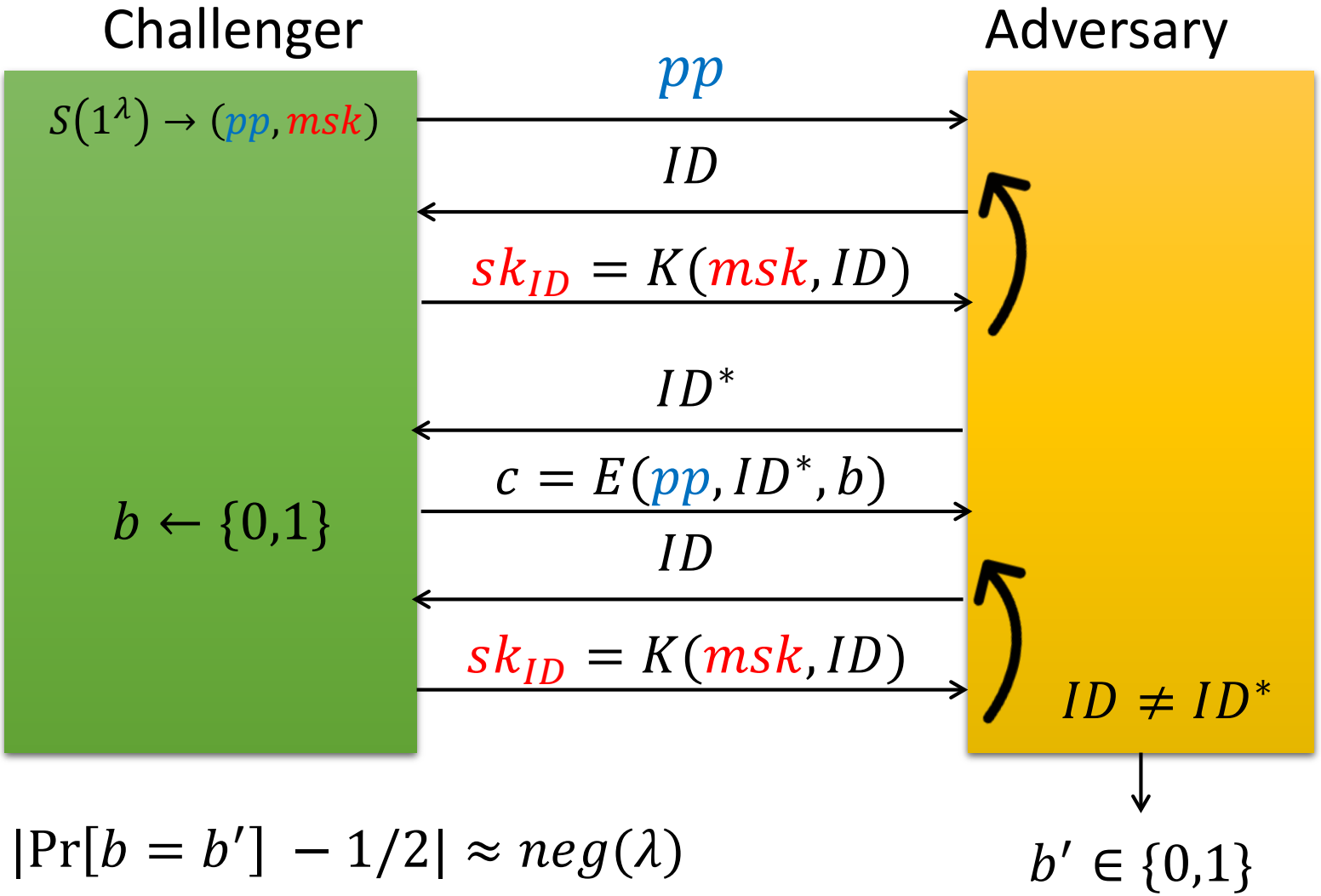
$D(sk_{ID}, c) \rightarrow m$ decrypt c using sk_{ID}

Full Security of IBE [BF01]

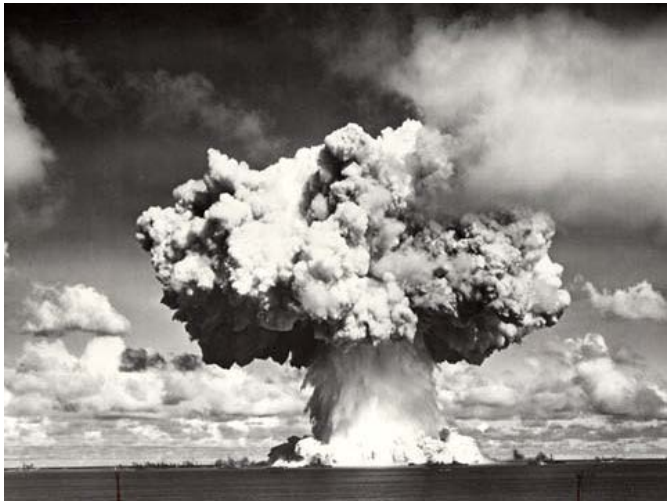


$|\Pr[b = b'] - 1/2| \approx neg(\lambda)$

Selective Security of IBE [CHK04]



$|\Pr[b = b'] - 1/2| \approx neg(\lambda)$



Can we realize IBE?

Yes, we can!
[Boneh and Franklin, CRYPTO 01]

Hierarchical IBE [HL02,GS02]

Use SK_{ID} to compute $SK_{ID|ID'}$ for any ID'

Bob

$SK_{bob@gmail.com}$ ⟶ $bob@gmail.com|April2,2018$

IBE Constructions

	Pairings	Lattices (LWE)	Quadratic Residuocity
IBE w/RO	BF01	GPV08	Cocks01 BGH07
IBE no RO	CHK03 BB04, W05 G06, W09	CHKP10 ABB10, MP12	??
HIBE	GS03, BB04…	CHKP10…	??

Can we realize IBE from weaker assumptions?

Negative Evidence

Trapdoor Permutations [BPRVW08]

Decisional Diffie-Hellman Assumption [PRV12]

ABE [SW05]

Hierarchical IBE

IBE

Reduce the Gap!

Public-key crypto

Public-Key Encryption

Trapdoor Functions

Private-key crypto

Signatures

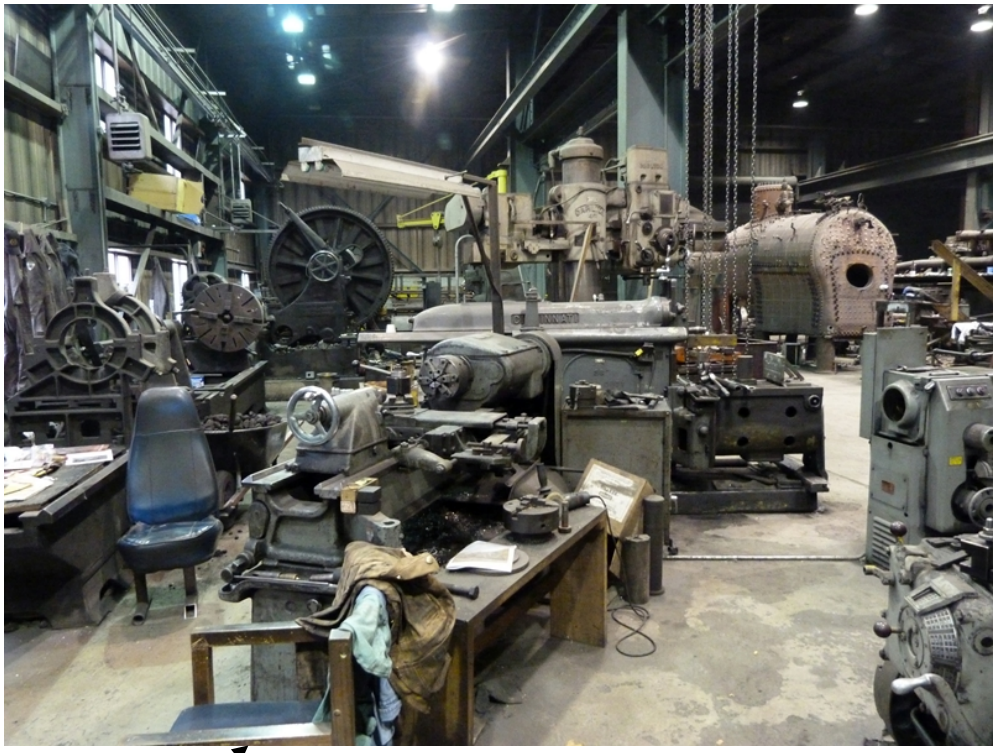
OWF PRG PRF

Our Results

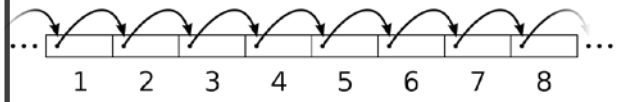
- Main result: IBE from Computational Diffie-Hellman Assumption (Fully-secure)
  - Or, the hardness of Factoring
- Selectively-Secure HIBE
  - In fact, from any IBE scheme!

Avoid impossibilities using non-black-box techniques.

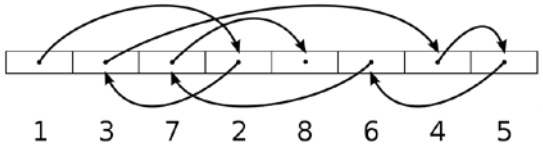
How do we get it?



Sequential access



Random access



Garbled RAM

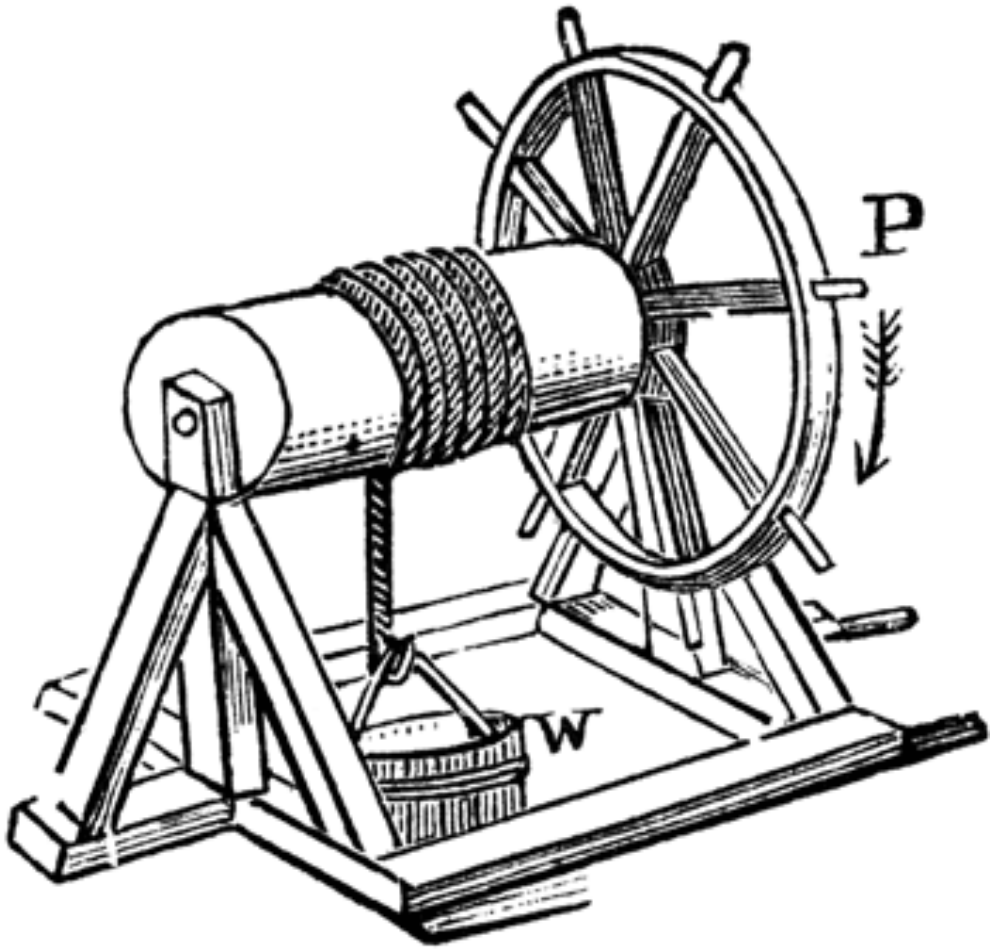
[LO13,GHLORW14,GLOS15,GLO15,GMP16, GGMP16,CDGGMP17]

```
int E,L,O,R,G[42][m],h[2][42][m],g[3][8],c
[42][42][2],f[42]; char d[42]; void v( int
b,int a,int j){ printf("\33[%d;%df\33[4%d"
"m  ",a,b,j); } void u(){ int T,e; n(42)o(
e,m)if(h[0][T][e]-h[1][T][e]){ v(e+4+e,T+2
,h[0][T][e]+1?h[0][T][e]:0); h[1][T][e]=h[
0][T][e]; } fflush(stdout); } void q(int l
                ,int k,int p){
                int T,e,a; L=0
                ; O=1; while(O
                ){ n(4&&L){ e=
                k+c[l] [T][0];
                h[0][L-1+c[l][
                T][l]][p?20-e:
```

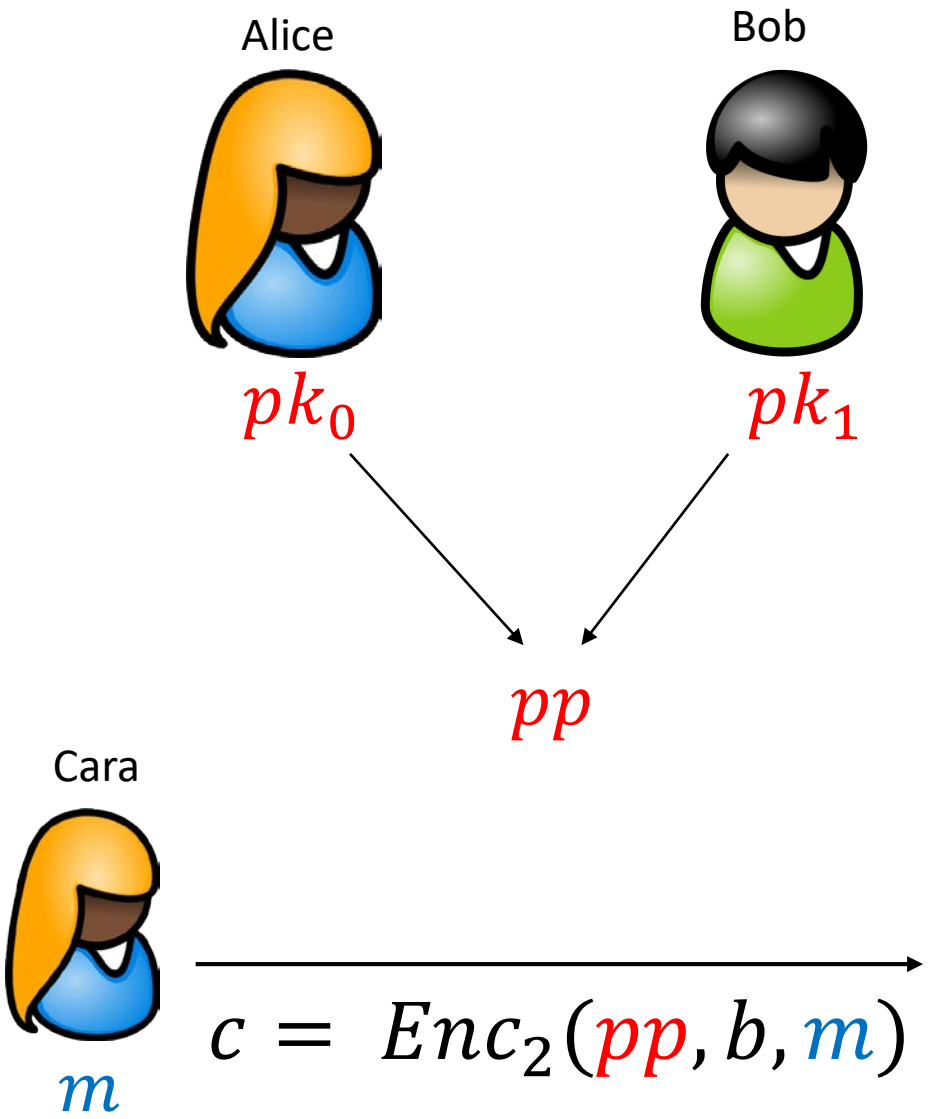
Witness Encryption

[CS00,GGSW13,BH15,CDGGMP17]

How do we get it?

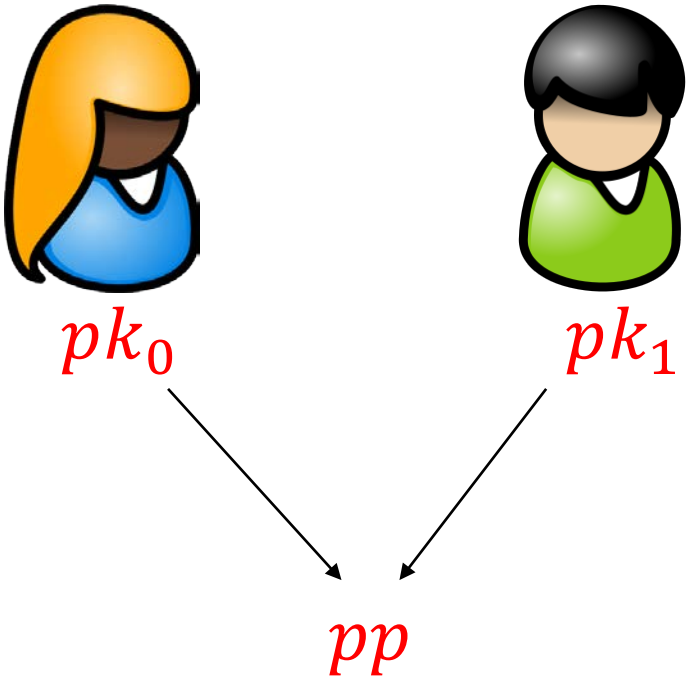


Compress two keys



- $|pp| = |pk_0| = |pk_1|$
- Encryption can be done to either pk_0 or pk_1 knowing just pp
- Decryption can be done using pk_0, pk_1 and the right secret key
- pp looses information about pk_0or pk_1

How known schemes from stronger assumptions compress two keys?



- pk_0 or pk_1 are correlated
- Structured assumptions
- Impossibility results: Similar intuition

Our goal: Compress Uncorrelated Keys!

Our Construction: Tools

Hash with Encryption

+

Yao's Garbled Circuits

Tool I: Hash with Encryption

Three Algorithms: (H, E, D)

$\mathrm{H}(x) \quad \rightarrow h$ $\quad$ h is short (say λ-bits)
x is 2λ-bits

Security: Hard to compute x, x' such that $H(x) = H(x')$

$E((h, i, b), m) \quad \rightarrow c$ $\quad$ where $i \in [2\lambda]$ and $b \in \{0,1\}$
$D(c, x) \quad \rightarrow m$ $\quad$ if $H(x) = h$ and $x_i = b$

Security: $x, E((h, i, 1 - x_i), 0) \approx x, E((h, i, 1 - x_i), 1)$

Reminiscent of Witness Encryption [GGSW13] or laconic OT [CDGGMP17].

Tool I: Hash with Encryption

Hash Parameters $\begin{pmatrix} A_{1,0} & A_{2,0} & \dots & A_{n,0} \\ A_{1,1} & A_{2,1} & \dots & A_{n,1} \end{pmatrix}$

- $\mathrm{H}(x) \to h$

$$h = \prod_{i \in [n]} A_{i,x_i}$$

Security can be argued based on DDH

$$g^x, g^y, g^{xy} \approx g^x, g^y, g^r$$

- $E((h,i,b),m) \to c = \left(\begin{matrix} A_{1,0}^s & A_{2,0}^s & \dots & A_{n,0}^s \\ A_{1,1}^s & A_{2,1}^s & \dots & A_{n,1}^s \end{matrix} \right.$  $\left. , \; h^s \oplus m \right)$

$A_{i,1-b}^s$

- $\mathrm{D}(c,x)$: Set $h^s = \prod_{i \in [n]} A_{i,x_i}^s$

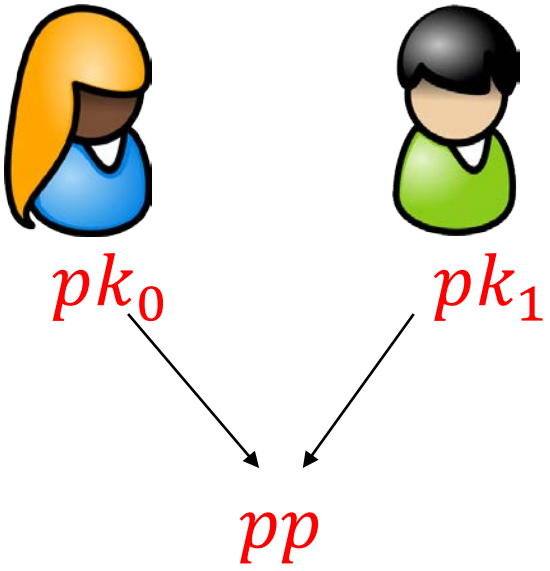
Tool 2: Yao's Garbled Circuits $(Garble, Eval)$ [Yao86, AIK04, AIK05, LP09, BHR12]

$Garble(C) \to \tilde{C}, \{lab_{i,0}, lab_{i,1}\}_i$

$Eval(\tilde{C}, lab_{i,x_i}) \to C(x)$

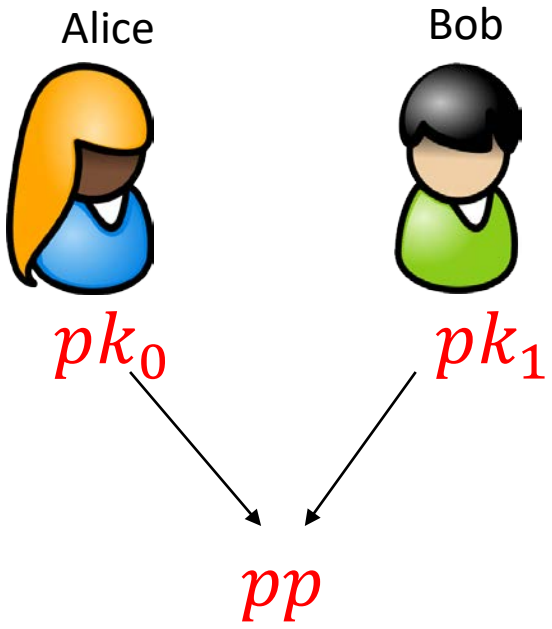
Security: $(\tilde{C}, lab_{i,x_i}) \approx Sim(C(x))$

How do we compress?



$$pp = H(pk_0|pk_1)$$

How do we encrypt?



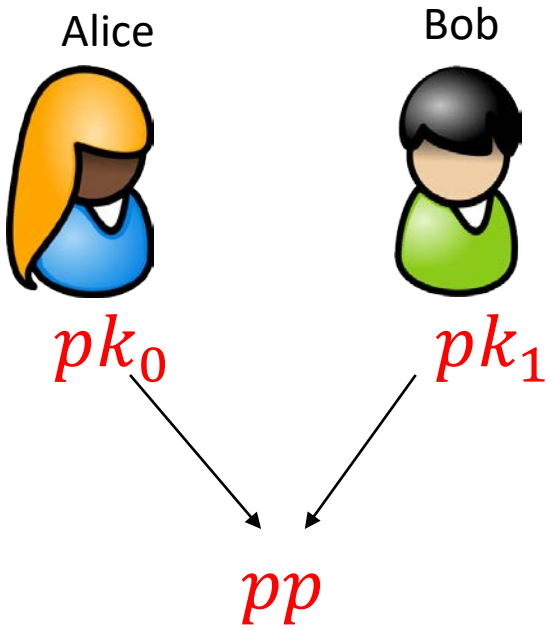
$$pp = H(pk_0|pk_1)$$

$P_{pp,b,m}(x)$

1. Abort if $pp \neq H(x)$.
2. If $b = 0$ then $pk = x[1 \ldots \lambda]$
   else $pk = x[\lambda + 1 \ldots 2\lambda]$
3. Output $Enc(pk, m)$

$$c = Enc_2(pp, b, m)$$

How do we encrypt?



$$pp = H(pk_0|pk_1)$$

$Enc_2(pp, b, m)$

- Circuit $C_m(pk) = Enc(pk, m)$
- $Garble(C_m) \rightarrow \tilde{C}, \{lab_{i,0}, lab_{i,1}\}_i$
- $\forall\, i \in \{b\lambda + 1, b\lambda + \lambda\}, \gamma \in \{0,1\}$
- $c_{i,\gamma} = E\left((pp, i, \gamma), lab_{i,\gamma}\right)$
- $c = \left\{\tilde{C}, \{c_{i,\gamma}\}\right\}$

$$c = Enc_2(pp, b, m)$$

How to decrypt?

- Decrypt $c = \left\{\tilde{C}, \{c_{i,\gamma}\}\right\}$ using pk_1, pk_2 and sk_b
- Recall $c_{1,0} = E\left((pp, b\lambda + 1, 0), lab_{1,0}\right)$ and

$$c_{1,1} = E\left((pp, b\lambda + 1, 1), lab_{1,1}\right)$$

  - which one can be decrypted?
  - $c_{1,pk_{b,1}}$ which decrypts to $lab_{1,pk_{b,1}}$
  - Similarly, for each i decrypt $c_{i,0}$ or $c_{i,1}$
- Evaluate($\tilde{C}$, $\{lab_{i,pk_{b,i}}\}$) outputs $Enc(pk_b, m)$

How to compress more keys/Bootstrapping?

- Using a Merkel Tree

- Exponentially Many Keys
  - Grow the tree dynamically – as needed

Chameleon Encryption

Five Algorithms: (S, H, H^{-1}, E, D)

$S(1^\lambda, n) \rightarrow (k, t)$ k is the hash Key

t is the hash trapdoor

$\mathrm{H}(k, x; r) \rightarrow h$ h is short (say λ-bits)

$\mathbf{H^{-1}}(\boldsymbol{t}, (\boldsymbol{x}, \boldsymbol{r}), \boldsymbol{x'}) \rightarrow \boldsymbol{r'}$ $\boldsymbol{H(k, x; r) = H(k, x'; r')}$

$E(k, (h, i, b), m) \rightarrow c$ where $i \in [n]$ and $b \in \{0,1\}$

$D(c, (x, r)) \rightarrow m$ if $H(k, x; r) = h$ and $x_i = b$

Security: $k, x, r, E(k, (h, i, 1 - x_i), 0) \approx k, x, r, E(k, (h, i, 1 - x_i), 1)$

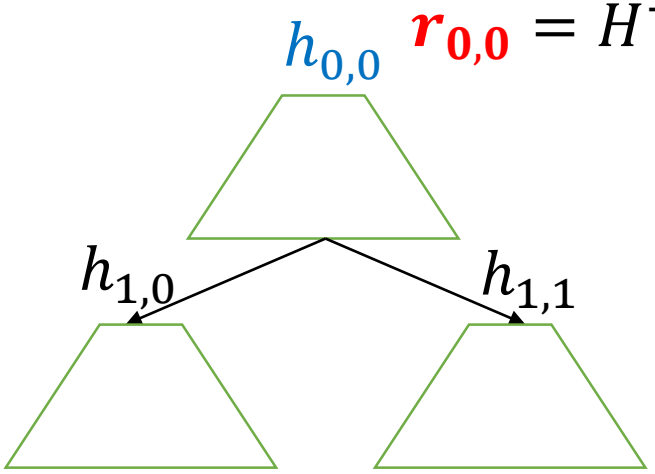
Bootstrapping

msk

t_0 k_0

t_1 k_1

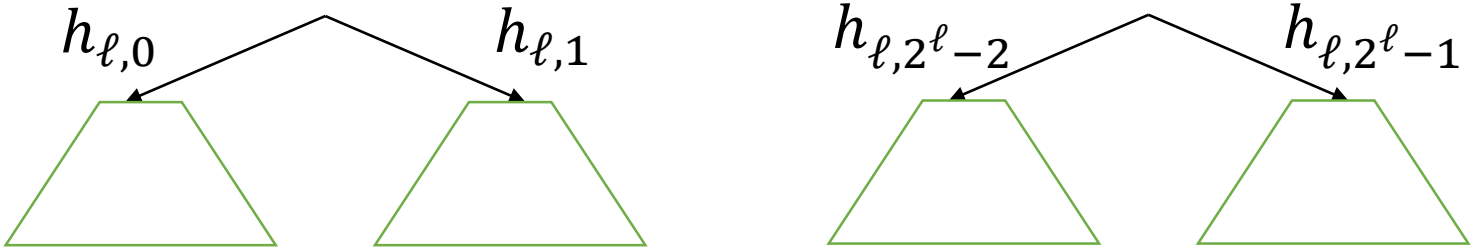
t_ℓ k_ℓ



$$\boldsymbol{r_{0,0}} = H^{-1}(t_0, (0^*, r'_{0,0}), h_{1,0} | h_{1,1})$$

$$h_{i,j} = H(k_i, 0^*; r'_{i,j})$$

$$\boldsymbol{r_{i,j}} = H^{-1}(t_i, (0^*, r'_{i,j}), h_{i+1,2j} | h_{i+1,2j+1})$$



Bootstrapping

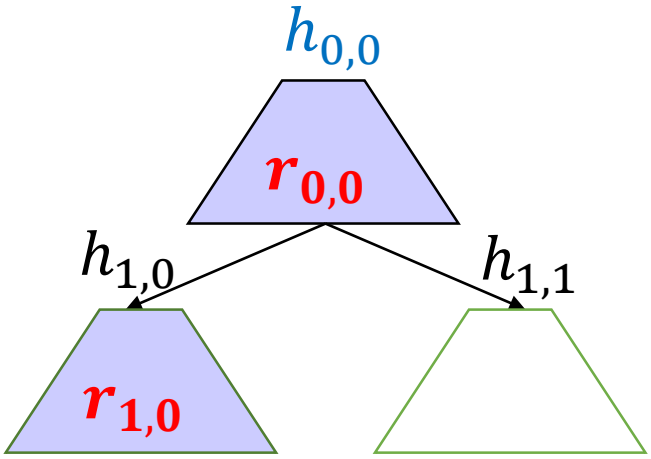
Secret-key for ID

msk

t_0 k_0

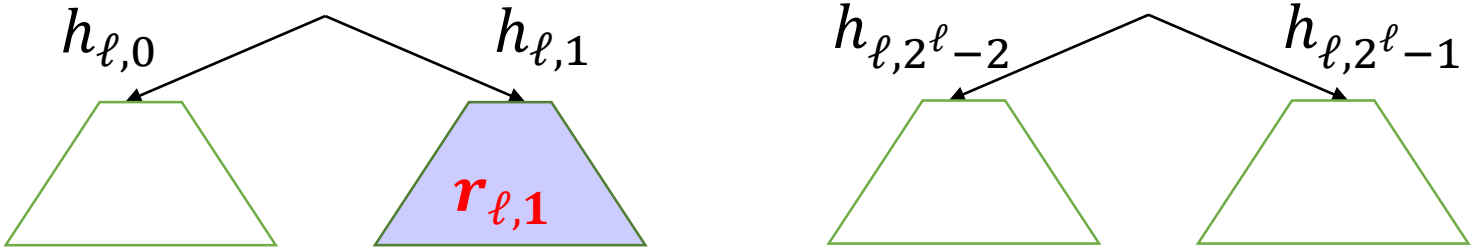
t_1 k_1

t_ℓ k_ℓ



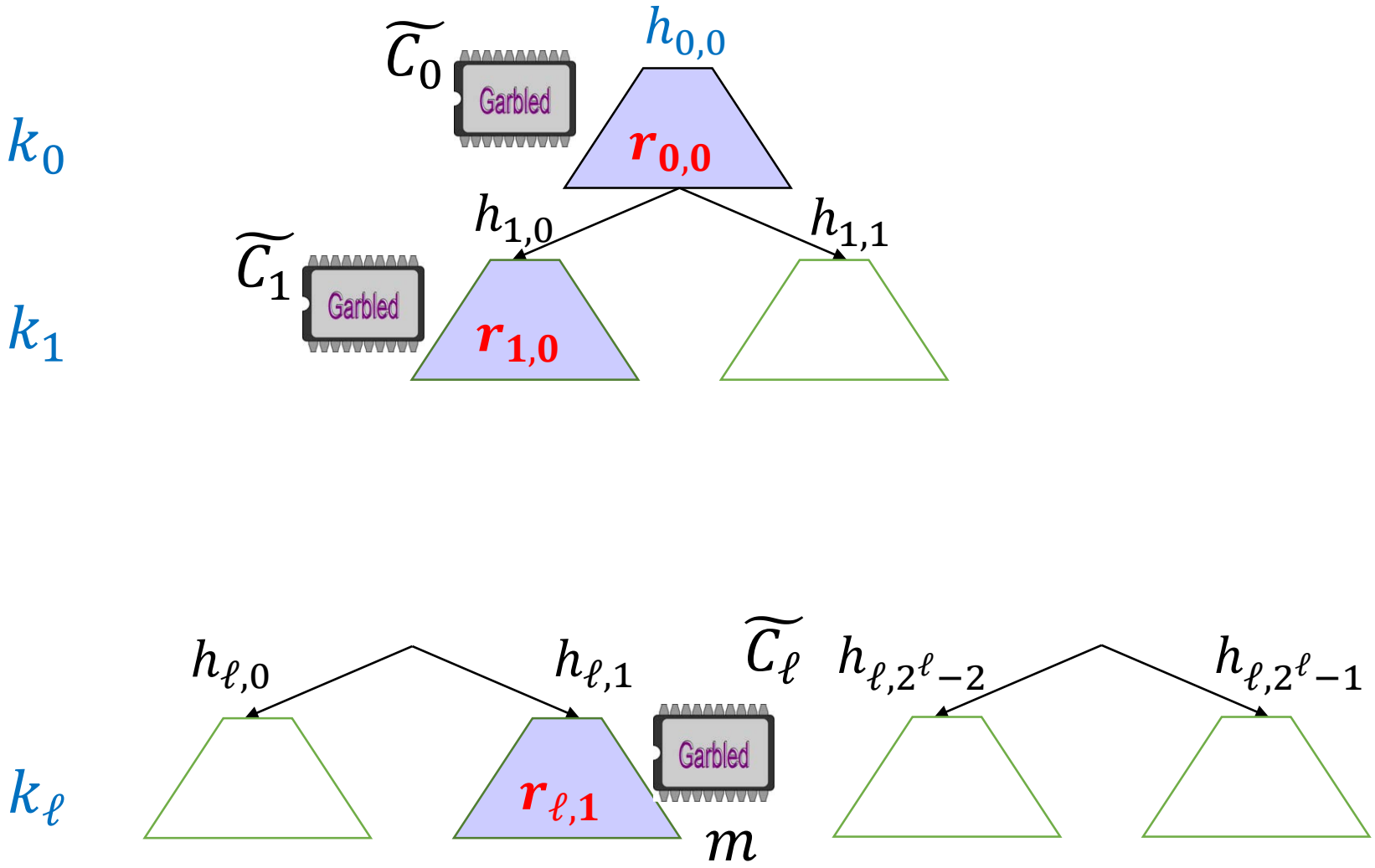
$$h_{i,j} = H(k_i, 0^*; r'_{i,j})$$

$$r_{i,j} = H^{-1}(t_i, (0^*, r'_{i,j}), h_{i+1,2j} | h_{i+1,2j+1})$$



Bootstrapping

Cipher for ID, m



Open Problems and Related Works

- Can we make the scheme efficient?
- IBE from any PKE?
- ABE from weaker assumptions?

- Techniques have other applications:
  - Laconic OT [CDGGMP17]
  - Anonymous IBE [BLSV18]
  - Circular Security [BLSV18,DGHM18,KT18]
  - Two-round MPC [GS17, GS18, BL18]
  - Adaptive garbled circuits/RAM [GS18a, GS18b]
  - Laconic Function Evaluation [QWW18]

Thank You! Questions?