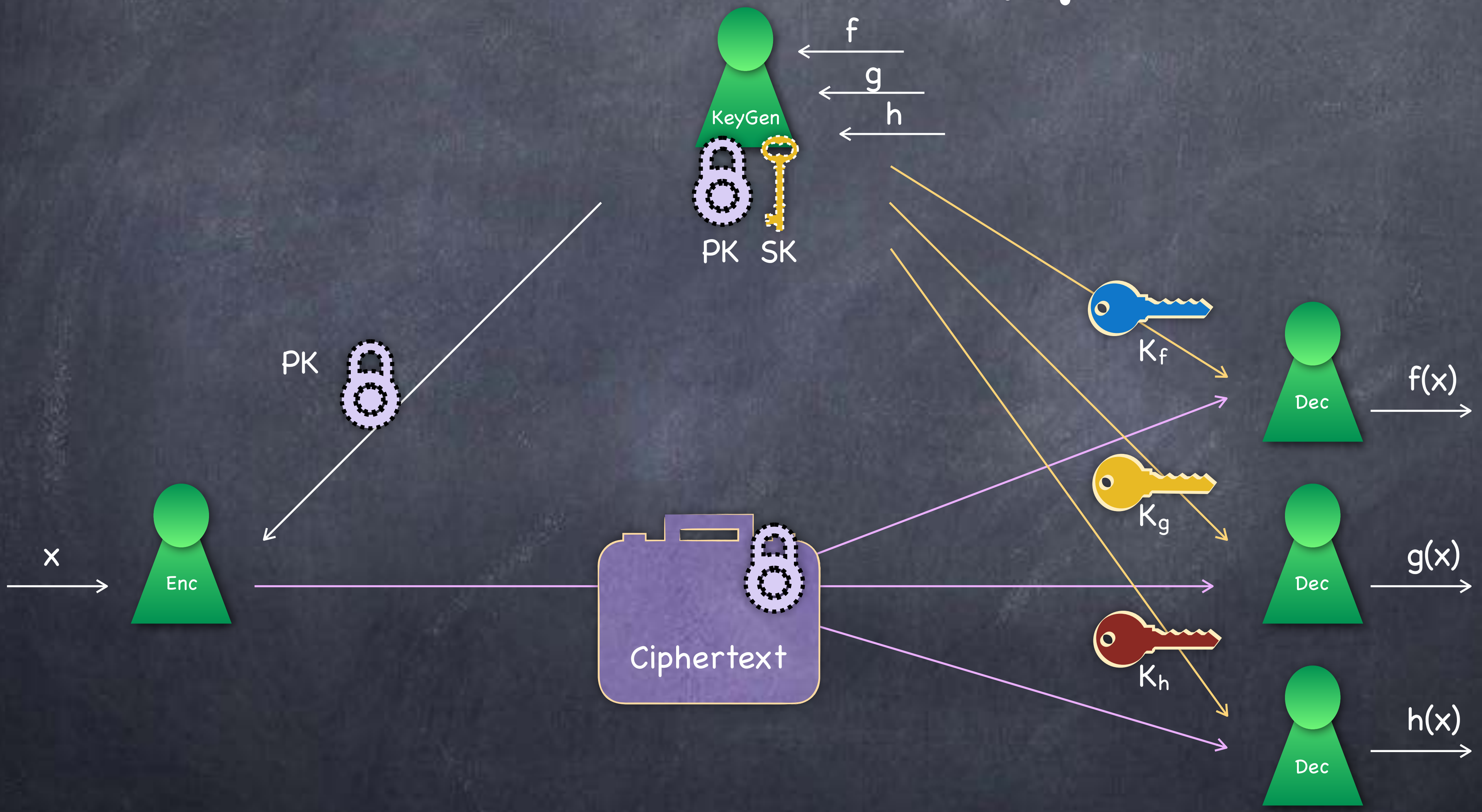
Functional Encryption

Lecture 22

Functional Encryption



Functional Encryption

- Key SK_f allows the decrypting party to learn f(x) from Enc(x)
- cf. FHE, can compute Enc(f(x)) from Enc(x), but cannot decrypt
- Obtaining multiple keys for f, g, h etc. should not let one learn more than f(x), g(x), h(x) etc.
  - Should not allow pooling keys to learn more information

Single-Key FE

- In which key for only one function will be ever be released
  - Function is not known when ciphertexts are created (otherwise trivial [Why?])
- A single-key FE scheme supporting arbitrary functions (with circuits of a priori bounded size)
  - Encryption of x is a Garbled circuit encoding the universal function: F(x,f) = f(x), with x being the garbler's input
  - Plus, 2n encrypted wire labels for the n input wires of f (using 2n public-keys in the master public-key)
  - Key for f: n secret-keys corresponding to the n bits of f
  - Can decrypt the labels of f → can evaluate F(x,f)

No Unbounded Sim-FE

- Suppose we require simulation-based security for FE
- Then there are function families which have no FE scheme that supports releasing an unbounded number of keys
- e.g., The message x is the seed of a PRF. The function f_z evaluates the PRF on the input z: $f_z(x) = PRF_x(z)$.
  - $\{ PRF_{x_j}(z_i) \mid j=1 \text{ to } N, i=1 \text{ to } N \}$ are N^2k-bit pseudorandom strings
  - Simulation should encode them into an $(LN+L'N)$-bit string (i.e., the simulated ciphertexts and keys)
    - If $Nk \gg L+L'$, not possible for truly random strings, and hence for pseudorandom strings too (even if simulator knows all z_i and all N^2k bits, but not any x_j, a priori)

Indistinguishability-Based FE

- (Weaker) Security definitions using a game between an adversary and a challenger
- Challenger gets (PK,SK) ← KeyGen, and gives PK to Adv
- Adv can ask for SK_f for any number of f of its choice
- Adv sends (m_0,m_1) to Challenger
- If $f(m_0)=f(m_1)$ for all f for which Adv received SK_f, Challenger picks b ← {0,1} and sends $Enc(m_b)$ to Adv
- Adv outputs b′ (as a guess for b)
- Security: ∀ PPT Adv, Pr[b′=b] ≈ ½
- Selective security: Adversary has to send (m_0,m_1) at first (before KeyGen is run)

Index-Payload Functions

- Message $x=(\alpha,m)$, and functions f_π s.t. $f_\pi(x)=(\alpha, m \text{ iff } \pi(\alpha)=1)$
  - α is the index which is public, and m is output iff $\pi(\alpha)=1$, where π is a predicate
  - Identity-Based Encryption (IBE): $\pi_\beta(\alpha) = 1$ iff $\alpha=\beta$
  - Attribute-Based Encryption (ABE)
    - Key-Policy ABE: $\alpha \in \{0,1\}^n$ and π a circuit (policy) over n Boolean variables
    - Ciphertext-Policy ABE: α a circuit (policy) over n Boolean variables, and π evaluates an input circuit on a fixed assignment
- Predicate Encryption: $x=(\alpha,m)$ and function f_π contains a predicate π s.t. $f_\pi(x) = m$ iff $\pi(\alpha)=1$ ($\perp$ otherwise).
  - Note: Not public-index, as α remains hidden

Identity-Based Encryption

- Identity-Based Encryption: $f_\beta(\alpha,m) = (\alpha,m)$ iff $\alpha=\beta$ (else $(\alpha,\perp)$)
- Useful as a public-key encryption scheme within an enterprise
- A key-server (with a master secret-key MSK and a master public-key PK) that can generate SK_{ID} for any given ID
  - Encryption will use PK, and the receiver's ID (e.g., email)
  - Receiver has to obtain SK_{ID} from the key-server

IBE from Pairing

- MPK: g,h, $Y=e(g,h)^y$, $\pi = (u,u_1,\ldots,u_n)$
- MSK: h^y
- Enc(m;ID) = (g^r, $\pi(ID)^r$, $m.Y^r$)

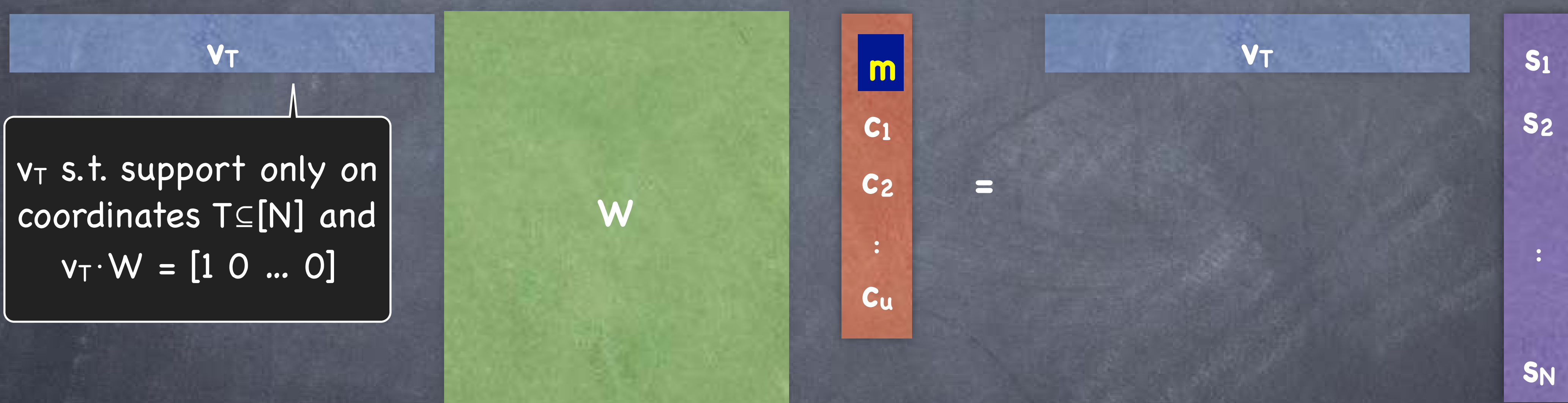
  $$\pi(ID) = u \prod_{i:ID_i=1} u_i$$
- SK for ID: (g^t, $h^y.\pi(ID)^t$) = (d_1, d_2)
- Dec (a, b, c; d_1, d_2) = c/ [$e(a,d_2)$ / $e(b,d_1)$]
- Full security based on Decisional-BDH

ABE schemes

- Easy solution for Single-Key CP-ABE, using secret-sharing
- The policy defines an access structure over the set of attributes
  - Secret-share the message for this access structure, and encrypt individual shares using attribute-specific keys PK_a
  - Key for an attribute set A, $SK_A = \{ SK_a \mid a \in A \}$
  - Note: cannot issue SK_A and $SK_{A'}$ as it allows computing $SK_{A \cup A'}$
- Will see how to use bilinear pairings for CP/KP-ABE to allow multiple keys when restricted to "linear policies"
  - Linear policies (a.k.a. Monotone Span Programs): the access structure (which sets of attributes allow decryption) is the access structure for a linear secret-sharing scheme

Linear Secret-Sharing

- Reconstruct($\sigma_{i_1},\ldots,\sigma_{i_t}$): pool together available coordinates $T \subseteq [N]$. Can reconstruct if there are enough equations to solve for m.

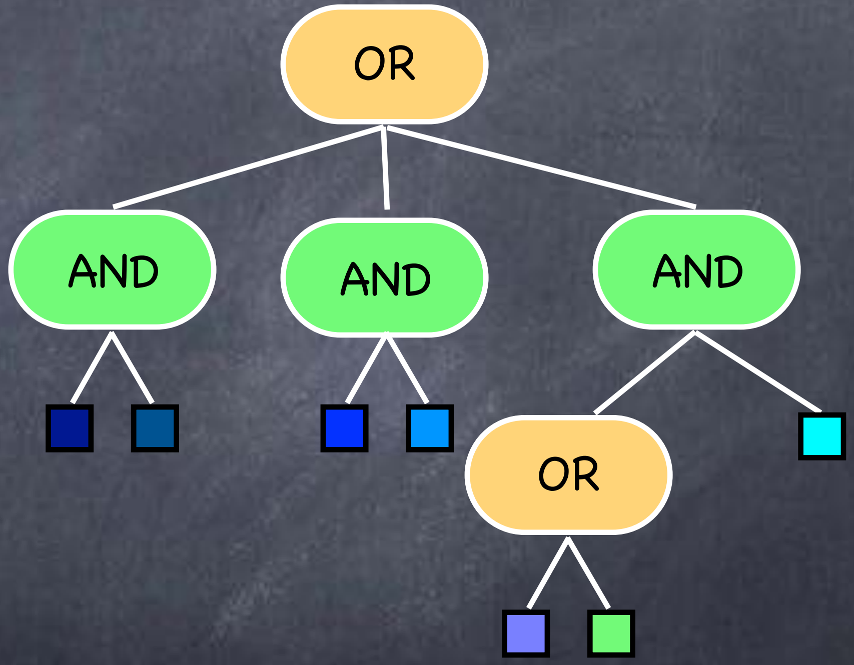


- Can work with any non-zero target vector **a** instead of [1 0 … 0] (by encoding m into **c** so that $\langle \mathbf{a},\mathbf{c} \rangle = m$)
  - [Exercise] An access structure has a linear secret-sharing scheme using [1 0 … 0] iff it has one with vector **a** (for any vector $\mathbf{a} \neq 0$)

Example of a Linear Policy

- Consider this policy, over 7 attributes
- W (with target vector [1 1 1 1]):

0	1	1	1
1	0	0	0
1	1	0	1
0	0	1	0
1	1	1	0
1	1	1	0
0	0	0	1



- Can generalize AND/OR to threshold gates

CP-ABE For Linear Policies

- PK: g, $Y=e(g,g)^y$, $Q=g^q$, $(T_1,\ldots,T_n) = (g^{t_1},\ldots, g^{t_n})$ (n attributes)
- MSK: g^y
- Enc(m,W;s,$r_1,\ldots,r_n$) = (W, $\{ Q^{\sigma_a} T_a^{-r_a} , g^{r_a} \}_{a\in[n]}$, g^s, $m.Y^s$) where $(\sigma_1,\ldots,\sigma_n)$ is a secret-sharing of s for access structure W
- SK for attribute set A: Let u be random. $SK_A = (K,L,\{ K_a \}_{a\in A})$ where $K=g^y.Q^u$, $L=g^u$, $K_a = T_a^u$
- Dec ($(W,\{Z_a,R_a\}_{a\in A},S,C)$; $(K,L,\{ K_a \}_{a\in A})$) : Get Y^s as $e(S,K)/ \prod_{a\in A} [e(Z_a,L)\cdot e(R_a,K_a)]^{v_a}$ where $v = [v_1 \ldots v_n]$ s.t. $v_a=0$ if $a \notin A$, and $v \underline{\sigma} = s$. Then $m = C/Y^s$
- Note: a random u for each key to prevent collusion
- Selective (attribute) security under strong assumptions

KP-ABE For Linear Policies

- PK: g, $Y=e(g,g)^y$, $T = (g^{t1},\ldots, g^{tn})$ (n attributes)
- MSK: y and t_a for each attribute a
- Enc(m,A;s) = (A, $\{ T_a^s \}_{a\in A}$, $m.Y^s$)
- SK for policy W (with n rows): Let $u=(u_1 \ldots u_n)$ s.t. $\Sigma_a u_a = y$. For each row a, let $x_a = \langle W_a,u\rangle/t_a$. Let Key $X = \{ g^{x_a} \}_{a\in[n]}$
- Dec ($(A,\{Z_a\}_{a\in A},C)$; $\{X_a\}_{a\in[n]}$) : Get $Y^s = \Pi_{a\in A}\ e(Z_a,X_i)^{v_a}$ where $v = [v_1 \ldots v_n]$ s.t. $v_a=0$ if $a \notin A$, and $v W = [1\ldots1]$. $m = C/Y^s$
- A random vector u for each key to prevent collusion
- Selective (attribute) security based on Decisional-BDH