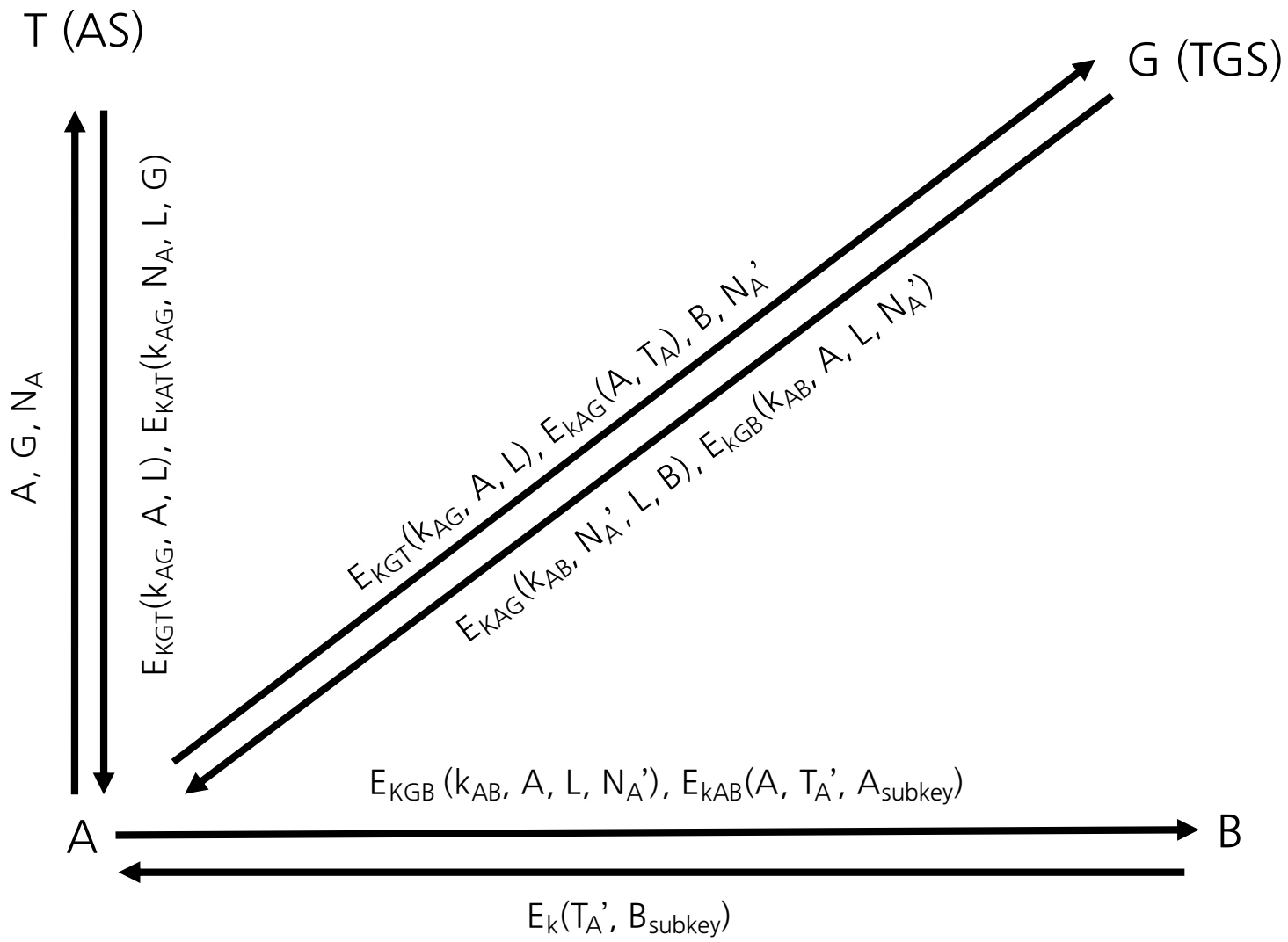
EE488
Introduction to
Cryptography Engineering

Yongdae Kim

Kerberos (scalable)



Combining PKE and DS

- Assurances of X.509 strong authentication
  - identity of A, and the token received by B was constructed by A
  - the token received by B was specifically intended for B;
  - the token received by B has “freshness”
  - the mutual secrecy of the transferred key.
- X.509 strong authentication
  - $D_A=(t_A, r_A, B, data_1, P_B(k_1))$, $D_B=(t_B, r_B, A, r_A, data_2, P_A(k_2))$,
  - $A \rightarrow B$: $cert_A$, D_A, $S_A(D_A)$
  - $B \rightarrow A$: $cert_B$, D_B, $S_B(D_B)$
- Comments
  - Since protocol does not specify inclusion of an identifier within the scope of the encryption P_B within D_A, one cannot guarantee that the signing party actually knows (or was the source of) plaintext key

Bilinear map and ID-based Encryption

$E_{kyd@cs.umn.edu}(m)_{???}$

Definition

- Bilinear Map
  - G_1 and G_2 be two abelian groups of prime order q.
  - additive notation for G_1: aP denotes the P added a times
  - the multiplicative notation for G_2
  - A map $e : G_1 \times G_1 \rightarrow G_2$ is called an admissible bilinear map if
    - » Bilinearity For any $P, Q \in G_1$ and $a, b \in Z_q$, $e(aP, bQ) = e(P, Q)^{ab}$
    - » Non-degeneracy $e(P, Q) \neq 1$ for at least one pair of $P, Q \in G_1$.
    - » Efficiency
- Hash functions
  - $h : \{0, 1\}^* \rightarrow \{0. 1\}^n$: A collision-free hash function
  - $H : \{0, 1\}^* \rightarrow G_1$: A collision-free full domain hash function (called map-to-point)
  - $H^*: G_2 \rightarrow Z_q$: A collision-free full domain hash function

Crypto Assumptions

- Playing with Bilinear maps
  - $e(aP, bQ) = e(P, abQ) = e(P, Q)^{ab}$
  - $e(aP, Q)\ e(cP, Q) = e(\ (a+c)\ P, Q)$
- Cryptographic Problems
  - DLP is hard on G_1 and G_2
    - » finding a from (P, aP) is hard
    - » finding a from $e(P, P)^a$ is hard
  - DDH is easy
    - » c = ab if and only if e(aP, bP) = e(cP, P).
  - BDHP is hard
    - » finding $e(P, P)^{abc}$ from aP, bP, cP is hard.

3-Way DH Key Agreement

- Let P be public generator of G1
- Three public keys: aP (Alice), bP (Bob), cP (Carol)
- Group key $G_{ABC}=e(P,P)^{abc}$
  - Alice computes $e(bP,cP)^{a}=e(P,P)^{abc}$
  - Bob computes $e(aP,cP)^{b}=e(P,P)^{abc}$
  - Carol computes $e(aP,bP)^{c}=e(P,P)^{abc}$
- Properties
  - No communication
  - Others cannot compute group key : BDH problem

Identity-Based Encryption

- ID=name+date of birth
- Trusted Third Party: secret s in Z_q
- Public params: generator P of G_1 and sP
- Secret Key Generation
  - ID_{Alice}: Alice → TTP
  - $sH(ID_{Alice})$: TTP → Alice
- Encryption: Bob encrypts for Alice
  - Pick random r in Z_q
  - Compute $g=e(H(ID_{Alice}), sP))$
  - Compute
    - » $g^r= e(H(ID_{Alice}), sP))^r= e(H(ID_{Alice}), rsP))= e(rH(ID_{Alice}), sP))$
  - Ciphertext: $\langle rP, c = m \text{ XOR } H_2(g^r) \rangle$

IBE (Cont'd)

- Decryption by Alice
  - Compute $g^r = e(H(ID_{Alice}), rsP)) = e(sH(ID_{Alice}), rP))$
  - Compute $H_2(g^r)$
  - $m = c$ XOR $H_2(g^r)$

- Why others cannot decrypt?
  - Others know only $H(ID_{Alice})$ and rP
  - It is hard to determine r from rP (DLP)
    - » thus they cannot compute g^r as $e(H(ID_{Alice}), sP))^r$
  - They don't know s
    - » cannot compute $e(H(ID_{Alice}), srP))$
  - They don't know $sH(ID_{Alice})$
    - » cannot compute $e(sH(ID_{Alice}), rP))$

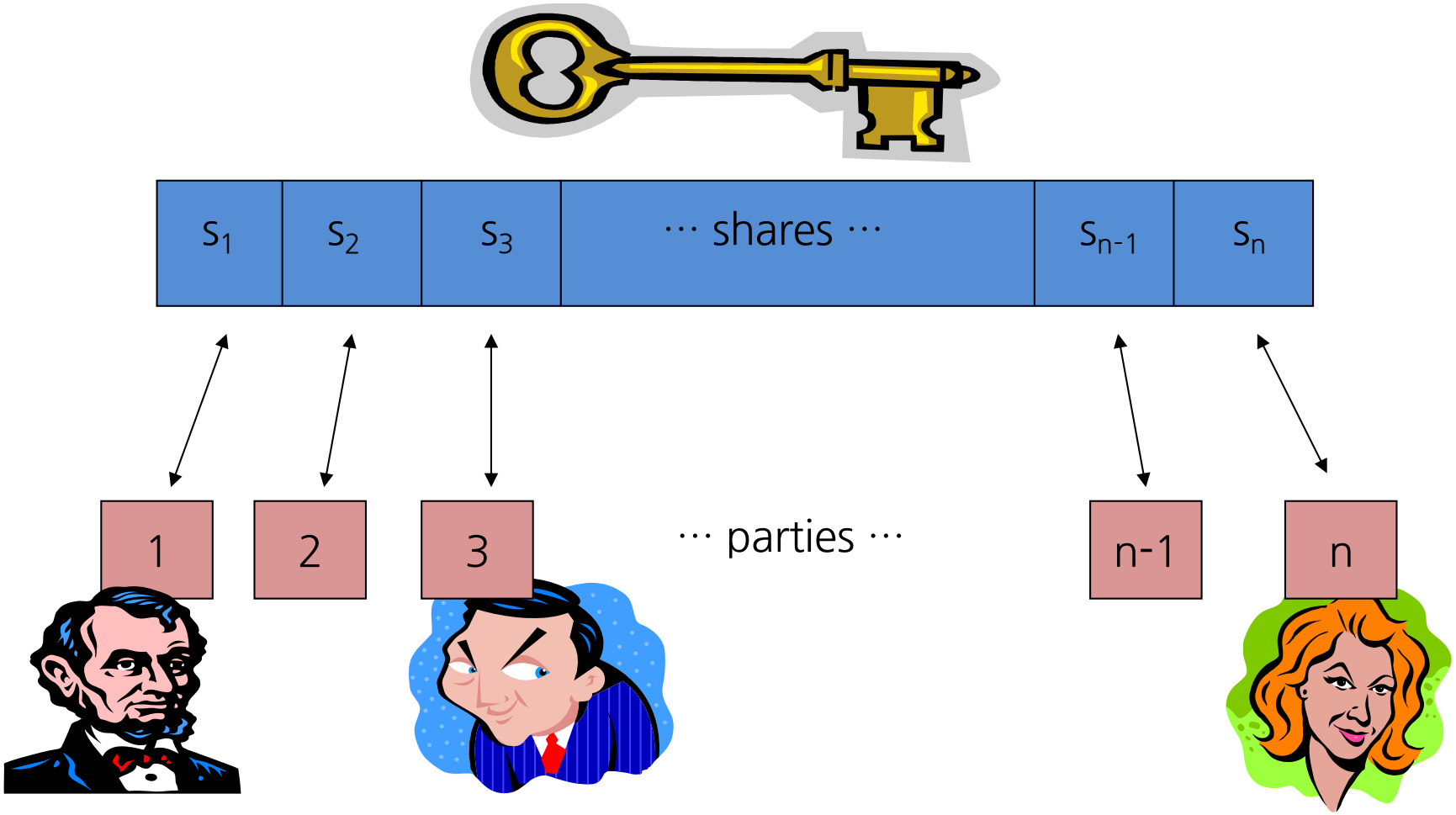
Discussion (PKI vs. Kerberos vs. IBE)

- On-line vs. off-line TTP
  - Implication?
- Non-reputation?
- Revocation?
- Scalability?
- Trust issue?

Threshold Crypto

- Motivating examples
  - (t, n) Threshold decryption
    - » A secret key K is encrypted by a public key of a group G.
    - » Each group member M_i knows a share SS_i of the group private key.
    - » When t members out of n group members get together, they can find the secret key.
  - (t, n) Threshold signature
    - » To be a member of an on-line community, you need signature from at least t board members out of total n board members.
    - » (t, n) threshold signature allows the member has a single certificate, which is computed from t partial certificates.

Conceptually…



Threshold Cryptography

- A group < threshold size t cannot determine the secret/perform the function
- A group >= threshold size t can always reconstruct the secret/perform the function
- Scheme will tolerate t-1 compromised/misbehaving parties
  - No information leakage when t-1 members get together!

(t,n) threshold scheme

- A polynomial *f*
  - degree t-1
- Dealer gives each party *i* secret $K_i = f(i)$
  - f(0) is the secret S.

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots + a_{t-1} x^{t-1} \pmod{p}$$

$$f(x) = \sum_{s=1}^{t} K_{\pi_B(s)} \prod_{j=1, j \neq s}^{t} \frac{(x - x_{\pi_B(j)})}{(x_{\pi_B(s)} - x_{\pi_B(j)})} \pmod{p}$$

  - $S = \Sigma_{i=1}^{t} c_i K_i$ where $c_i = \prod_{1 \le j \le t,\ j \neq i} x_j / (x_j - x_i)$

(t, n) Threshold ElGamal

- Encryption
  - generate random integer k and compute $r = g^k \bmod p$
  - compute $c = m\, y^k \bmod p$
  - Ciphertext (r, c)
- Decryption
  - $m = c\, r^{-a} \bmod p$
- Threshold decryption
  - Note that private key $a = \Sigma_{i=1}^{t} c_i K_i$ $(c_i = \prod_{1 \le j \le t, j \neq i} x_j/(x_j - x_i))$
  - So 1) t members compute r^{Ki} 2) raise it to c_i to get r^{ciKi} and 3) mutiply all of them to get r^a.
- Threshold DSA Signature is similar…

How to prevent break-ins

- As time goes by more and more board members could be corrupted (or compromised)!
- Change shares but not the secret
- f'(x) = f(x) + g(x) where g(0) = 0.
- f'(0) = S still.
- Attacker who compromises t-1 within the refresh interval has no information.
  - SS_i will be changed to f'(i).

Questions?

- Yongdae Kim
  - email: yongdaek@kaist.ac.kr
  - Home: http://syssec.kaist.ac.kr/~yongdaek
  - Facebook: https://www.facebook.com/y0ngdaek
  - Twitter: https://twitter.com/yongdaek
  - Google "Yongdae Kim"