Tutorial – Ring Signature

CSCI361 – Computer Security

Sionggo Japit
sjapit@uow.edu.au

10 February 2020

Ring Signatures

- A Ring signature is a digital signature that is created by a member of a group of trusted set of signers, who each of them has his/her own key.
- The scheme works in such a way that it is not possible to determine the identify of the signer after the signer sign a document.
- Ring signature scheme was initially created by Ron Riverst, Adi Shamir, and Yael Tauman in 2001.

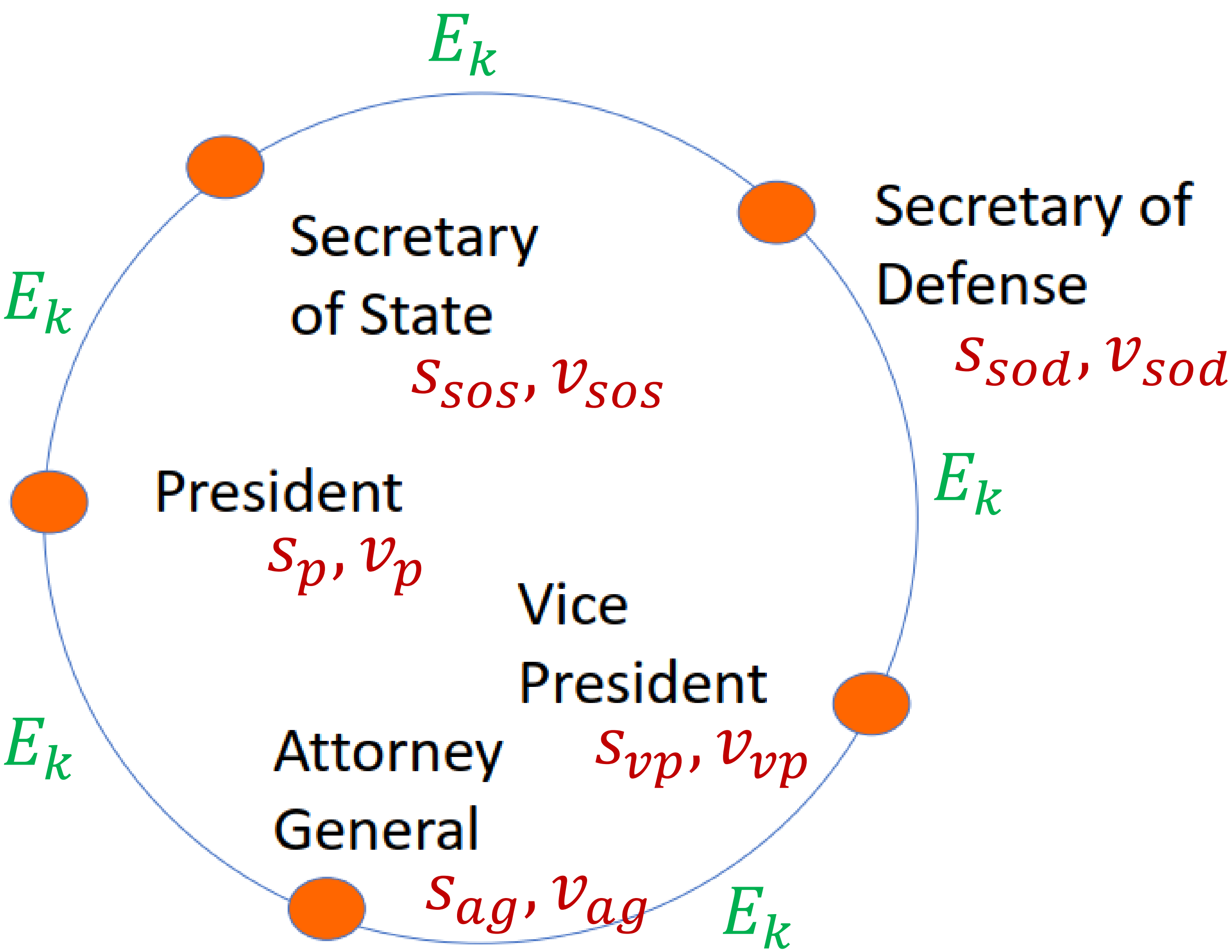
Ring Signatures

- In a ring signature, all trusted signers (entities) uses public key crypto system; that is, all have their own public key and private key.
- Public keys are publicly known to every body in the trusted set (group), and private keys are kept secret to each individual signer.
- If signer i wants to sign a message $(message)$, for example, the signer will use his/her own private key s_i, but the public keys of the others signers in the group $(v_1, v_2, \dots v_n)$.

Ring Signatures

- Once the message is signed, it should then be possible to check the validity of the group by knowing the public keys of signers in the group, but it is not possible to determine the individual who had signed the message because the private key used are kept secret.

Ring Signatures



Ring Signatures

The steps:

1. Generate encryption with $k = \text{Hash}(message)$.
2. Generate a random value (u).
3. Encrypt u to give $v = E_k(u)$.
4. For each person in the group, apart from the signer:
   - Calculate $e = {s_i}^{v_i} \ (mod\ N_i)$, where s_i is the random number generated for the private key (disguised/faked signing key) of the i^{th} person, and v_i is the public key (verifying key).

Ring Signatures

Step 1 – Key setup

1. Using RSA (for example), generate public and private keys for all participants.
2. Randomly choose a glue value v.
3. Compute a key for an encryption system: $k = H(m)$.

Ring Signatures

Step 2 – Generating the ring signature

1. For all participants, except the signer, compute values $y_i = x_i^{e_i} \, mod \, n_i$, where
    - y_i is the computed private key (signing key) of i^{th} member, (Note, this private key (signing key) is the key to be used to produce the ring signature, it is not the same as the private key of the member determined in Step 1.)
    - x_i is the random number generated for the computation of the private key (y_i) member, and
    - e_i is the public key of i^{th} member.

Ring Signatures

2. Solve the ring equation to calculate the private key (signing key) of the signer. (Note: this private key (signing key) is the signing key for the generation of ring signature, it is not the same as the private key determined in Step 1.)

   The ring equation is:

   $v = E_k\left(y_s \oplus E_k(y_i \oplus v)\right)$, where

   - $1 \leq i \leq n$, and $i \neq s$
   - y_s is the private key (signing key) of the ring signature signer.

Ring Signatures

$$v = E_k\left(y_s \oplus E_k(y_i \oplus v)\right)$$

$$E_k^{-1}(v) = \left(y_s \oplus E_k(y_i \oplus v)\right)$$

$$y_s = E_k^{-1}(v) \oplus E_k(y_i \oplus v)\ mod\ n_s$$

3. Once y_s is computed, the combination function of ring equation can be realized, that is,

   $$\text{combinedFunction} = E_k\left(y_s \oplus E_k(y_i \oplus v)\right)$$

Ring Signatures

4. Compute x_s of the signer (producer) of the ring signature:

   $x_s = (y_s)^{d_s}\ mod\ n_s$, where

   - y_s is the signing key of ring signature signer.
   - d_s is the actual private key of the signer (determined in Step 1.)

5. The ring signature generated is:
   $\left((e_1)(e_2)\ldots(e_n), v, x_1, x_2, \ldots, x_n\right)$

Ring Signatures

Step 3 – Verification

1. Compute $y_i = x_i^{e_i} \, mod \, n_i$ for all members of the group.
2. Compute the key for an encryption system: $k = H(m)$.
3. Solve the ring equation and verify that the combined function $E_k\left(y_s \oplus E_k(y_i \oplus v)\right)$ is matching. If the combined function is not matching with the once obtained in Step 2, the verification fail.

Ring Signatures

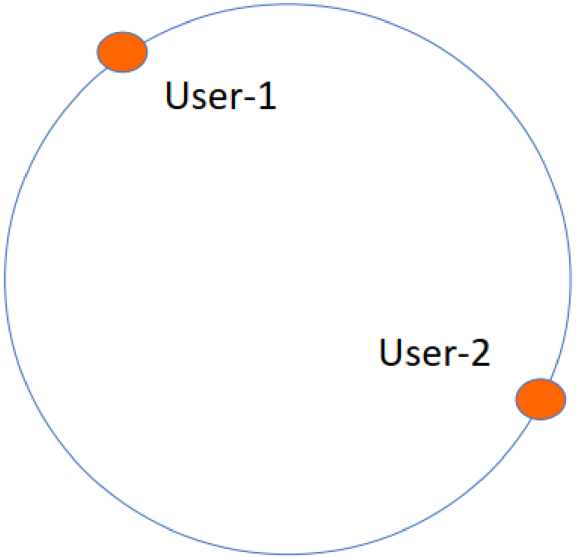
- For example, Alice (user-1) and Bob (User-2) form a group, and each generate their public and private key.

  Alice public key: (5, 21)

  Alice private key: (5, 21)

  Bob public key: (3, 33)

  Bob private key: (7, 33)
- Bob wish to generate a ring signature.
- The steps are shown next:



========== Ring Signature ==========
Step 1: ==========> Setup
Alice public key: (5, 21)
Alice private key: (5, 21)

Bob public key: (3, 33)
Bob private key: (7, 33)

Random glue value: 69

The message: ringsignature

The key for TEA algorithm (Symmetric encryption used in this example:
key 1: 4121642264
key 2: 1157955899
key 3: 1949256732
key 4: 4242056894

key for symmetric encryption (Hash(message)):
f5ab45184505013b742f4c1cfcd8a6be1538be0f83ab78e7ea81e04e14f02c57

Step 2: ==========> Ring signature generation
Assuming Bob is to generate a ring signature....
Bob computes xa and ya for Alice:
xa : 4, ya: 16

Bob solves the ring equation E_k(yb XOR E_k(ya XOR glue)) = glue
Bob rearrange the ring equation to yb = D_k(glue) XOR E_k(ya XOR glue)
Bob first computes ya XOR glue: 85
Bob then computes encrypted yaXORGlue: -7096328078488358901
Bob next computes decrypted glue: -1525309384793490721
Bob obtains yb = 24
Bob next computes the combinedFunction1 (yb XOR E_k(ya XOR glue)): -19
Bob then computes xb using yb and his private key: 18
xb: 18
The generated ring signature: ((5, 21)(3, 33), 69, 4, 18)

Step 3: ==========> Verification
Verifier computes ya and yb:b
ya: 16
yb: 24
Verifier checks E_k(yb XOR E_k(ya XOR glue) = combinedFunction2
Verifier computes ya XOR Glue: 85
Verifier computes encrypted ya XOR Glue: -7096328078488358901
Verifier next calculate the combine function: -19

combinedFunction1 = -19 is the same as combinedFunction2 = -19
Verification of ring signature is successful!