Multi-key Hierarchical Identity-Based Signatures

Hoon Wei Lim

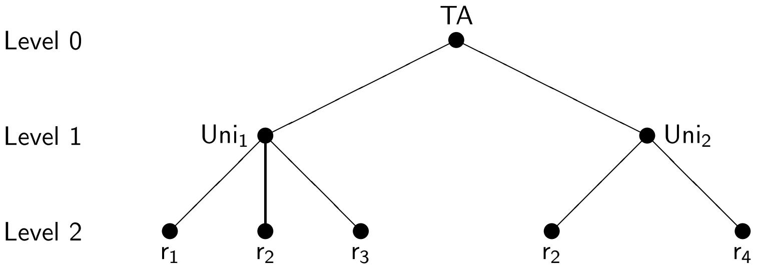
Nanyang Technological University

9 June 2010

Outline

Role-based access control:

- *Role signatures* based on hierarchical identity-based signatures (HIBS):
  - using signing keys associated with role identifiers;
  - hierarchical namespace.
- Let Alice have roles r_1=`lecturer`, r_2= `professor` and r_3= `IEEE member`.
- If Alice wants to access some restricted documents using roles r_1 and r_3:
  - principle of least privilege;
  - then she signs a request using the corresponding private keys.

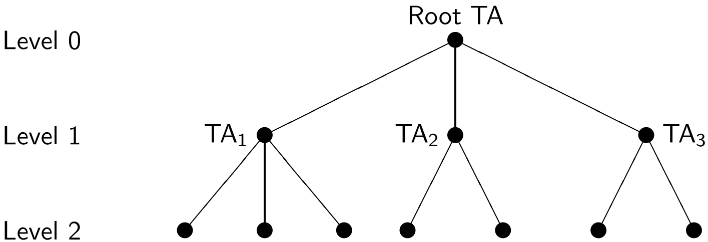


TA
Level 0
Level 1
Level 2
Uni1
Uni2
r1
r2
r3
r2
r4

Mobile ad hoc networks (MANETs):

- Use of identity-based cryptography is attractive:
  - avoids certificate management;
  - meets low bandwidth requirement.
- Nodes may be compromised or unavailable:
  - so it is desirable to distribute the function of a trusted authority (TA) across multiple nodes.
- Nodes can obtain multiple private keys from multiple TAs:
  - private keys are then aggregated when used for signing.

Introduction
Motivating examples



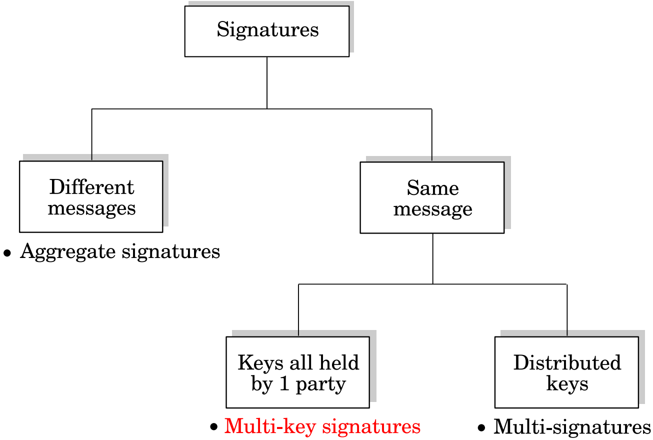
Question to be answered:

How do we *efficiently* and *securely* aggregate a set of private keys when signing a message?

- The essence of our new primitive, i.e. *multi-key signatures*:
  - based on hierarchical identity-based cryptography;
  - user owns multiple identifiers and thus possesses a set of corresponding private keys;
  - a single signature is produced using a combination of multiple private keys on a selected message;
  - identifiers may be located at arbitrary positions in the hierarchy.

- Identity-based multi-signatures [Gentry-Ramzan'06, Bellare-Neven'07]:
  - a set of users all sign the same message;
  - non-interactive and interactive.
- Identity-based aggregate signatures [Gentry-Ramzan'06]:
  - a set of users each signs a different message;
  - non-interactive (but requires coordination of state).
- Identity-based threshold signatures [Baek-Zheng'04]:
  - t (threshold) out of n parties first compute individual shares, which are then combined into a single signature;
  - interactive.
- Differences from multi-key HIBS: **efficiency, security, flexibility**.

Introduction
Related concepts



- Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups where $|\mathbb{G}| = |\mathbb{G}_T| = q$, a large prime, then an admissible pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has properties:
  - *Bilinear*: Given $P, Q, R \in \mathbb{G}_1$, we have

$$\begin{aligned} \hat{e}(P, Q + R) &= \hat{e}(P, Q) \cdot \hat{e}(P, R) \text{ and} \\ \hat{e}(P + Q, R) &= \hat{e}(P, R) \cdot \hat{e}(Q, R). \end{aligned}$$

    Hence, for any $a, b \in \mathbb{Z}_q^*$, we have

$$\begin{aligned} \hat{e}(aP, bQ) &= \hat{e}(abP, Q) = \hat{e}(P, abQ) \\ &= \hat{e}(aP, Q)^b = \hat{e}(P, Q)^{ab}. \end{aligned}$$

  - *non-degeneracy*: $e(P, P) \neq 1$ for some $P \in \mathbb{G}$.
  - *computability*: $e(P, Q)$ can be efficiently computed.

Computational Diffie-Hellman (CDH) problem in $\mathbb{G}$:

Given $\langle P, aP, bP \rangle \in \mathbb{G}$ for some random $P \in \mathbb{G}$ and randomly chosen $a, b \in \mathbb{Z}_q^*$, compute $abP \in \mathbb{G}$.

Multi-key HIBS
Definition

- Root Setup: It generates the system parameters and a master secret on input a security parameter λ.
- Lower-level Setup: It picks a secret value to be used to issue private keys to lower-level children.
- Extract: An entity with identifier $\mathrm{ID}_t = id_1, \ldots, id_t$ computes a private key S_{t+1} for any of its children with identifier $\mathrm{ID}_{t+1} = id_1, \ldots, id_t, id_{t+1}$.

Multi-key HIBS

Definition

- $\textsc{Sign}$: Given a set $\mathsf{SK} = \{S^j_{t_j} : 1 \leq j \leq n\}$ of private keys, a message M, and the system parameters, this algorithm outputs a signature σ.
- $\textsc{Verify}$: Given a signature $\sigma \in \mathcal{S}$, a set $\mathsf{ID} = \{\mathsf{ID}^j_{t_j} : 1 \leq j \leq n\}$ of identifiers, a message M, and the system parameters, this algorithm outputs `valid` or `invalid`.

- Consistency: $\textsc{Verify}(\textsc{Sign}(\mathsf{SK}, M), \mathsf{ID}, M) = \texttt{valid}$.

Multi-key HIBS

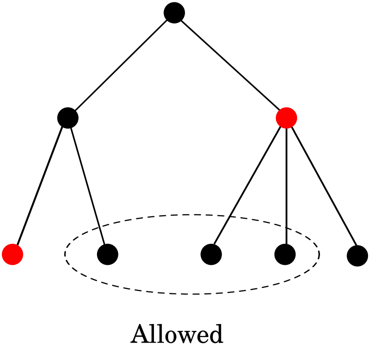
Security model

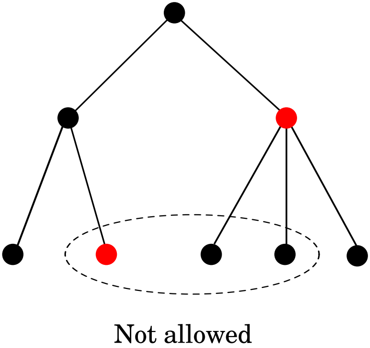
Extend the normal HIBS security game [Gentry-Silverberg'02]:

- Challenger runs $\textsc{Root Setup}$ and adversary $\mathcal{A}$ is given the system parameters.
- $\mathcal{A}$ is given access to extract and sign oracles.
- $\mathcal{A}$ outputs a forgery σ^*, a set of target identifiers ID^*, and a message M^*.
- $\mathcal{A}$ wins the game if the following are *all* true:
  - $\textsc{Verify}(\sigma^*, \mathsf{ID}^*, M^*) = \texttt{valid}$;
  - The adversary has not made a sign query on input ID^*, M^*;
  - There exists an identifier $\mathsf{ID}' \in \mathsf{ID}^*$ for which the adversary has not made an extract query on ID' or any of its ancestors.

Multi-key HIBS

Security model





● Compromised node

Multi-key HIBS
Construction

Main idea:

- Adaptation of the Gentry-Silverberg HIBS scheme:
  - re-use of the ROOT SETUP, LOWER-LEVEL SETUP and EXTRACT algorithms.
- When signing:
  - arrange identifiers in lexicographic order;
  - private key components are summed before generating a normal HIBS.
- For verification:
  - extend the VERIFY algorithm of the Gentry-Silverberg scheme.

Multi-key HIBS

Construction

- Root Setup: The root Private Key Generator (PKG)
  - generates $\mathbb{G}$ and $\mathbb{G}_T$ of prime order q and an admissible pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ on input λ;
  - chooses a generator $P_0 \in \mathbb{G}$;
  - picks a random value $s_0 \in \mathbb{Z}_q^*$ and sets $Q_0 = s_0 P_0$;
  - selects cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}$;
  - sets the master secret to be s_0 and the system parameters $\langle \mathbb{G}, \mathbb{G}_T, e, q, P_0, Q_0, H_1, H_2 \rangle$.
- Lower-level Setup: A lower-level entity (lower-level PKG or user) at level $t \geq 1$ picks a random secret $s_t \in \mathbb{Z}_q^*$.

Multi-key HIBS

Construction

- EXTRACT: For an entity with identifier $\text{ID}_t = id_1, \ldots, id_t$, the entity's parent:
  - computes $P_t = H_1(\text{ID}_t) \in \mathbb{G}$;
  - sets $S_t = \sum_{i=1}^{t} s_{i-1} P_i = S_{t-1} + s_{t-1} P_t$;
  - defines $Q_i = s_i P_0$ for $1 \leq i \leq t-1$;
  - private key $\langle S_t, Q_1, \ldots, Q_{t-1} \rangle$ is given to the entity by its parent.
- Note that up to this point, our scheme is identical to the Gentry-Silverberg HIBS scheme.

- SIGN: Given any $n \geq 1$ and a set $\mathsf{SK} = \{\langle S^j_{t_j}, Q^j_1, \ldots, Q^j_{t_j-1}\rangle : 1 \leq j \leq n\}$ of n private keys associated with a set $\mathsf{ID} = \{\mathsf{ID}^j_{t_j} : 1 \leq j \leq n\}$ of identifiers, and a message M, the signer:
  - chooses a secret value $s_\varphi \in \mathbb{Z}^*_q$;
  - computes $P_M = H_2(\mathsf{ID}^1_{t_1}, \ldots, \mathsf{ID}^n_{t_n}, M)$;
  - calculates

$$\varphi = \sum_{j=1}^{n} S^j_{t_j} + s_\varphi P_M \quad \text{and} \quad Q_\varphi = s_\varphi P_0;$$

  - outputs the signature $\sigma = \langle \varphi, \mathsf{Q}, Q_\varphi \rangle$, where $\mathsf{Q} = \{Q^j_i : 1 \leq i \leq t_j - 1, 1 \leq j \leq n\}$.

- VERIFY: Given $\sigma = \langle \varphi, \mathrm{Q}, Q_\varphi \rangle$, a set of identifiers $\mathrm{ID} = \{\mathrm{ID}^1_{t_1}, \ldots, \mathrm{ID}^n_{t_n}\}$ and a message M, the verifier:
  - computes $P^j_i = H_1(\mathrm{ID}^j_i)$ for $1 \leq i \leq t_j$ and $1 \leq j \leq n$;
  - computes $P_M = H_2(\mathrm{ID}^1_{t_1}, \ldots, \mathrm{ID}^n_{t_n}, M)$;
  - checks if $e(P_0, \varphi)$ is equal to

$$\left(\prod_{j=1}^{n} \prod_{i=1}^{t_j} e(Q^j_{i-1}, P^j_i) \right) \cdot e(Q_\varphi, P_M),$$

  outputting `valid` if this equation holds, and `invalid` otherwise.

Security Analysis

- We first look at the security of our multi-key IBS (1-level multi-key HIBS) scheme.
- Our security proof is in the Random Oracle Model.
- We extend proof techniques used for the Boneh-Franklin IBE scheme.

Security Analysis

Theorem

Suppose that $\mathcal{A}$ is a forger against our multi-key IBS scheme that has success probability ϵ. Then there is an algorithm $\mathcal{B}$ which solves the CDH problem in groups $\mathbb{G}$ equipped with a pairing, with advantage at least

$$\epsilon/(\mathbf{e} \cdot q_{H_1} \cdot q_{H_2}).$$

Security Analysis

Proof techniques:

- Based on interactions between algorithms $\mathcal{A}$ (forger) and $\mathcal{B}$ (simulator);
- $\mathcal{B}$ generates the system parameters and embeds an instance of the CDH problem;
- $\mathcal{A}$ submits queries to $\mathcal{B}$;
- $\mathcal{B}$ injects an instance of the CDH problem in one randomly chosen response to a H_1 query:
  - so that $\mathcal{A}$'s forgery may help $\mathcal{B}$ solve the CDH problem;
- $\mathcal{B}$ controls the relevant oracles and must either respond correctly or abort.

Security Analysis

Proof techniques for the more complicated multi-key HIBS scheme:

- Borrow Gentry-Silverberg's simulation techniques for handling H_1 and extract queries in the hierarchical setting:
  - $\mathcal{B}$ randomly injects an instance of the CDH problem into responses to H_1 queries.
- Combine the above techniques with our approach to handling sign queries, and obtain a security reduction.
- However, so far we have only obtained a security proof for some special cases:
  - constructing a proof for the general case remains an open problem.

Discussion

Efficiency comparison

	ADD	eMUL	PAI	HASH	mMUL	EXP
Bellare-Neven IBMS						
signing	-	-	-	$n(n+1)$	n^2+n-1	$2n$
verification	-	-	-	$n-1$	n	2
Gentry-Ramzan IBMS						
signing	$3n-2$	$2n$	0	n	-	-
verification	$n-1$	0	3	$n+1$	-	-
Multi-key IBS						
signing	n	2	0	1	-	-
verification	$n-1$	0	3	$n+1$	-	-

- Main saving – signing cost!

Discussion

Reducing verification cost

- Our verification algorithm can be optimised in special cases, when identifiers are:
  - at the same level, and have a common parent;
  - at the same level, but have different parents;
  - at different levels, but have a common ancestor;
- Having common ancestors indicate common Q-values and public keys, thus certain pairing computations can be eliminated.

$$e(P_0, \varphi) = \left(\prod_{j=1}^{n} \prod_{i=1}^{t_j} e(Q_{i-1}^{j}, P_i^{j}) \right) \cdot e(Q_\varphi, P_M)$$

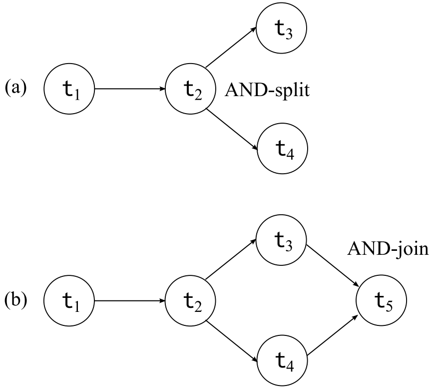
Discussion

Extension

- From hierarchical to *workflow signatures*:
  - reflecting workflow logical relationships, such as AND-join and AND-split.
  - providing proofs of workflow compliance, reflecting the sequence of task execution and the relevant logical relationships.
- Modification to the multi-key HIBS scheme:
  - the EXTRACT algorithm may now take as input multiple private keys.

Discussion

Extension



Open Problems

- Constant size signatures – potentially more efficient verification.
- Instantiation in the standard model.
- Generalisation of multi-key HIBS to the threshold setting:
  - demonstrate knowledge of a subset of size t of a set of private keys of size n.
- Construction in the normal (non-identity-based) public key setting:
  - perhaps by adapting the BGLS aggregate signature scheme.

Acknowledgement

- Joint work with Professor Kenny Paterson, Royal Holloway, University of London.
- Research was funded by the UK EPSRC under grant EP/D051878/1.