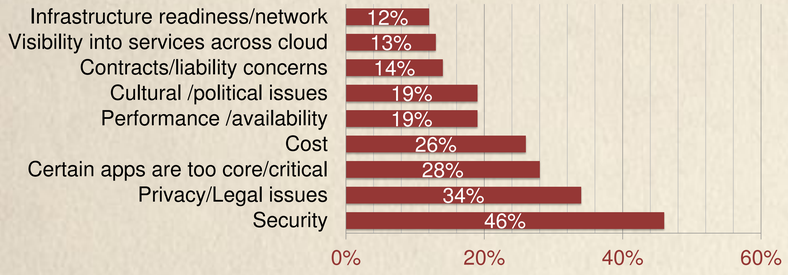
Protection of Sensitive Data in Clouds Using Active Privacy Bundles and Agent-Based Secure Multiparty Computation

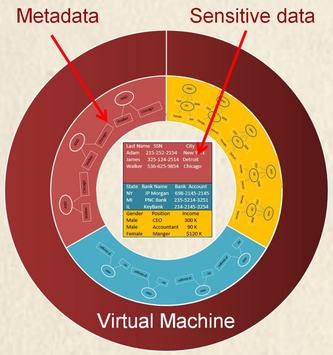
Akram Y. Sarhan (Advisor: Prof. Leszek T. Lilien)
Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008

Introduction

- Challenges for protecting data in clouds (cf. TechInsights Report, 2013)
  - „Security" below includes privacy



- Privacy and security challenges in clouds (Jansen, 2011)
  - Data leakage, performance, risk management, efficient data storage
- Two types of solutions for cloud-based privacy and security
  1) Centralized solutions
     - Rely on centralized *trusted third parties* (*TTPs*)
  2) Decentralized solutions
     - Avoid relying on centralized TTPs
- Problems with using TTPs
  - Bottleneck, insecure, single point of failure
- *Active Privacy Bundle* (*APB*)
  - Sensitive data: user data
  - Metadata: describes various *policies* for sensitive data
    - Dissemination control policy
    - Access control policy
    - Integrity self-check specification
  - *Virtual machine* (*VM*): executes APB, incl. three privacy/security activities
    - Integrity self-checking
    - *Evaporation*: *partial* APB self-destruction
    - *Apoptosis*: *complete* APB self-destruction

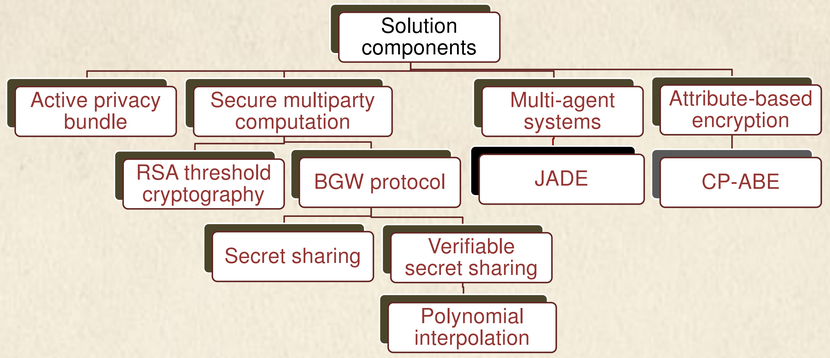


Motivation and Objectives

- Providing adequate privacy and security for data in clouds
  - Self-protecting data
  - Fine-grained access control
  - Fault tolerance
- Protect cloud data against attackers
  - Dishonest cloud providers
  - Unauthorized sub-contractors
  - Dishonest tenants (i.e.,.other cloud users)
- Protecting data with *decentralized* TTP (without *centralized* TTP)
  - Using *multi-agent systems* (*MAS*) for implementating decentralized TTPs
  - Using MAS for performance improvements
    - Thanks to *parallel processing* of data

Methods

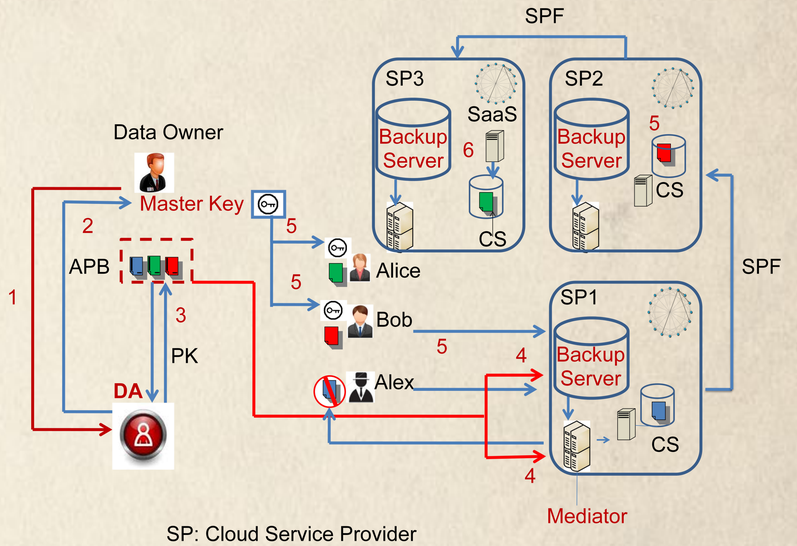
- Solution components and their roles



- *Active privacy bundle* (*APB*)
  - Encapsulates and protects sensitive data throughout their full lifecycle
  - Protects against tampering, privacy violations, unauthorized access or dissemination
- *Secure multiparty computation* (*SMC*)
  - Multiple parties can *jointly* compute some value, based on individually held *secret* inputs or functions
    - While assuring privacy of their secrets to one another in the process
  - *RSA threshold cryptography*
    - Several parties (more than a threshold number) must cooperate to encrypt/decrypt data
  - *BGW* (*Ben-Or, Goldwasser and Widgerson*) *protocol*
    - Used to jointly compute a chosen function *f* on shared or private input
- *Multi-agent systems* (*MAS*)
  - Distributed computing with *intelligent* multiple agents —with JADE implementation
- *Attribute-based encryption* (*ABE*)
  - One-to-many encryption scheme based on *public key*
  - *Ciphertext-policy attribute-based encryption* (CP-ABE)
    - Private key uses ABE and cipher-text specifies an access policy over an attribute set

Results: The Proposed Solution

- Major results
  - Designed and partially developed the APB-SMC scheme
    - Integrated SMC into APB implementation
      - SMC uses RSA threshold cryptography and BGW protocol
    - APB-SMC replaces the *centralized* TTP with a *distributed* trust mediator
    - SMC used in *constructing* and *enabling APB*
    - Enhanced APB evaporation
    - Enhanced APB apoptosis
  - Integrated ABE and CP-ABE into APB-SMC
    - Provide higher *security* and *fault tolerance*
    - Support *access right delegation* and *revocation*
- APB creation and enabling algorithms in APB-SMC
  1) APB creation
     - Identify sensitive data
     - Create access policy attributes
     - Create access structure
     - Generate public and master keys
     - Encrypt sensitive data
     - Encrypt metadata
     - Hash and sign the APB
     - Encrypt APB
     - Plan APB itinerary
  1) *APB creation – cont.*
     - Send APB to a cloud backup server
     - Store APB copy on the backup server
     - Send APB to cloud mediator
     - Send APB according to its itinerary
  2) APB enabling
     - APB host trust verification
     - APB permission
     - APB integrity verification
     - APB policy enforcement
     - APB decryption
- Using APB-SMC to protect sensitive data in clouds



SP: Cloud Service Provider
DA: Decentralized Authority
SPF: Service Provider Federation
CS: Cloud Storage

The activities identified by numbers in the above figure:

1) Owner requests DA to encrypt the data outsourced to the cloud
2) DA generates the public keys (PK) and master keys
3) DA encrypts the APB using the encryption algorithm that takes as its input the outsourced data, PK, and an access structure
4) APB visits two proxies available in every cloud supporting APB-SMC: (i) a *backup APB server*; (ii) a (decentralized) *mediator server*
5) A user is authorized to access cloud data and services if he received private keys from the data owner (if his access was not revoked)
6) Additional data protection possible (e.g., *homomorphic encryption*)

Conclusions

- Current work status
  - Completed design of the APB-SMC scheme
  - Working on modeling, formal model analysis, simulation experiments
- Future work
  - Demonstrate that APB-SMC provides privacy, security, fault tolerance and efficiency
  - Integrate a multi-agent system (MAS) framework into APB-SMC
  - Validate and optimize MAS-based APB-SMC