Attribute-Based Access Control

insider threats, security, privacy and trust

Vladimir Oleshchuk

Universitetet i Agder
Norway

Outline

Access Control

- Mechanism that ensures only authorized users have access to computer resources
- The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner
- The logical component that serves to receive the access request from the subject, to decide, and to enforce the access decision.

Access Control

- Mechanism that ensures only authorized users have access to computer resources
- The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.
- The logical component that serves to receive the access request from the subject, to decide, and to enforce the access decision.

Access Control

- Mechanism that ensures only authorized users have access to computer resources
- The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.
- The logical component that serves to receive the access request from the subject, to decide, and to enforce the access decision.

Access Control

- Mechanism that ensures only authorized users have access to computer resources
- The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.
- The logical component that serves to receive the access request from the subject, to decide, and to enforce the access decision.

Insiders: informal definitions

- Someone with access, privilege, or knowledge of information systems and services. (Brackney and Anderson)
- Anyone operating inside the security perimeter. (Patzakis)
- Wholly or partially trusted subject.
- A system user who can misuse privileges.
- Someone with legitimate past or present access to resources.
- Someone with authorized access who might attempt unauthorized removal or sabotage of critical assets or who can aid outsiders in doing so.

Insiders: informal definitions

1. Someone with access, privilege, or knowledge of information systems and services. (Brackney and Anderson)
2. Anyone operating inside the security perimeter. (Patzakis)
3. Wholly or partially trusted subject.
4. A system user who can misuse privileges.
5. Someone with legitimate past or present access to resources.
6. Someone with authorized access who might attempt unauthorized removal or sabotage of critical assets or who can aid outsiders in doing so.

Insiders: informal definitions

1. Someone with access, privilege, or knowledge of information systems and services. (Brackney and Anderson)
2. Anyone operating inside the security perimeter. (Patzakis)
3. Wholly or partially trusted subject.
4. A system user who can misuse privileges.
5. Someone with legitimate past or present access to resources.
6. Someone with authorized access who might attempt unauthorized removal or sabotage of critical assets or who can aid outsiders in doing so.

Insiders: informal definitions

1. Someone with access, privilege, or knowledge of information systems and services. (Brackney and Anderson)
2. Anyone operating inside the security perimeter. (Patzakis)
3. Wholly or partially trusted subject.
4. A system user who can misuse privileges.
5. Someone with legitimate past or present access to resources.
6. Someone with authorized access who might attempt unauthorized removal or sabotage of critical assets or who can aid outsiders in doing so.

Insiders: informal definitions

1. Someone with access, privilege, or knowledge of information systems and services. (Brackney and Anderson)
2. Anyone operating inside the security perimeter. (Patzakis)
3. Wholly or partially trusted subject.
4. A system user who can misuse privileges.
5. Someone with legitimate past or present access to resources.
6. Someone with authorized access who might attempt unauthorized removal or sabotage of critical assets or who can aid outsiders in doing so.

Insiders: informal definitions

1. Someone with access, privilege, or knowledge of information systems and services. (Brackney and Anderson)
2. Anyone operating inside the security perimeter. (Patzakis)
3. Wholly or partially trusted subject.
4. A system user who can misuse privileges.
5. Someone with legitimate past or present access to resources.
6. Someone with authorized access who might attempt unauthorized removal or sabotage of critical assets or who can aid outsiders in doing so.

Insiders: informal definitions

1. Someone with access, privilege, or knowledge of information systems and services. (Brackney and Anderson)
2. Anyone operating inside the security perimeter. (Patzakis)
3. Wholly or partially trusted subject.
4. A system user who can misuse privileges.
5. Someone with legitimate past or present access to resources.
6. Someone with authorized access who might attempt unauthorized removal or sabotage of critical assets or who can aid outsiders in doing so.

Trust and insider threats

- Trust is assumption about the way an entity will behave
- Authorized users (trusted) are assumed not to abuse their privileges for which they are authorized (trusted).
- Insider threat: when a trusted authorized user is not trustworthy
- Insider threats $\Longrightarrow$ abuse of privileges

Definition

The insider problem arises because the set of users trusted on some level in some particular context is not equal to the set of trustworthy users on the same level for the same context.

Trust and insider threats

- Trust is assumption about the way an entity will behave
- Authorized users (trusted) are assumed not to abuse their privileges for which they are authorized (trusted).
- Insider threat: when a trusted authorized user is not trustworthy.
- Insider threats $\Longrightarrow$ abuse of privileges.

Definition

The insider problem arises because the set of users trusted on some level in some particular context is not equal to the set of trustworthy users on the same level for the same context.

Trust and insider threats

- Trust is assumption about the way an entity will behave
- Authorized users (trusted) are assumed not to abuse their privileges for which they are authorized (trusted).
- Insider threat: when a trusted authorized user is not trustworthy.
- Insider threats $\Longrightarrow$ abuse of privileges.

Definition

The insider problem arises because the set of users trusted on some level in some particular context is not equal to the set of trustworthy users on the same level for the same context.

Trust and insider threats

- Trust is assumption about the way an entity will behave
- Authorized users (trusted) are assumed not to abuse their privileges for which they are authorized (trusted).
- Insider threat: when a trusted authorized user is not trustworthy.
- Insider threats $\Longrightarrow$ abuse of privileges.

Definition

The insider problem arises because the set of users trusted on some level in some particular context is not equal to the set of trustworthy users on the same level for the same context.

Trust and insider threats

- Trust is assumption about the way an entity will behave
- Authorized users (trusted) are assumed not to abuse their privileges for which they are authorized (trusted).
- Insider threat: when a trusted authorized user is not trustworthy.
- Insider threats $\Longrightarrow$ abuse of privileges.

Definition

The insider problem arises because the set of users trusted on some level in some particular context is not equal to the set of trustworthy users on the same level for the same context.

Trust and insider threats

- Trust is assumption about the way an entity will behave
- Authorized users (trusted) are assumed not to abuse their privileges for which they are authorized (trusted).
- Insider threat: when a trusted authorized user is not trustworthy.
- Insider threats $\Longrightarrow$ abuse of privileges.

Definition

The insider problem arises because the set of users trusted on some level in some particular context is not equal to the set of trustworthy users on the same level for the same context.

Trust and insider threats

- Trust is assumption about the way an entity will behave
- Authorized users (trusted) are assumed not to abuse their privileges for which they are authorized (trusted).
- Insider threat: when a trusted authorized user is not trustworthy.
- Insider threats $\Longrightarrow$ abuse of privileges.

Definition

The insider problem arises because the set of users trusted on some level in some particular context is not equal to the set of trustworthy users on the same level for the same context.

Insider problem:

How abuse of privileges (legally granted by access control services) can be effectively discovered, prevented and mitigated (countered) ?

Insider problem:

How abuse of privileges (legally granted by access control services) can be effectively discovered, prevented and mitigated (countered) ?

Prevention

To counter insider threats and minimize mismatch (inconsistency) between trust and trustworthiness following features should be supported:

1. Using of proper authentications methods (some are more secure then other);
2. Management of privileges should not be based on identities;
3. Support of finer levels of granularity;
4. Contextual-awareness;
5. Support dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy, trust-awareness).
6. Provide dynamic policies combining access control and risk.

Prevention

To counter insider threats and minimize mismatch (inconsistency) between trust and trustworthiness following features should be supported:

1. Using of proper authentications methods (some are more secure then other);
2. Management of privileges should not be based on identities;
3. Support of finer levels of granularity;
4. Contextual-awareness;
5. Support dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy, trust-awareness).
6. Provide dynamic policies combining access control and risk.

Prevention

To counter insider threats and minimize mismatch (inconsistency) between trust and trustworthiness following features should be supported:

1. Using of proper authentications methods (some are more secure then other);
2. Management of privileges should not be based on identities;
3. Support of finer levels of granularity;
4. Contextual-awareness;
5. Support dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy, trust-awareness).
6. Provide dynamic policies combining access control and risk.

Prevention

To counter insider threats and minimize mismatch (inconsistency) between trust and trustworthiness following features should be supported:

1. Using of proper authentications methods (some are more secure then other);
2. Management of privileges should not be based on identities;
3. Support of finer levels of granularity;
4. Contextual-awareness;
5. Support dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy, trust-awareness).
6. Provide dynamic policies combining access control and risk.

Prevention

To counter insider threats and minimize mismatch (inconsistency) between trust and trustworthiness following features should be supported:

1. Using of proper authentications methods (some are more secure then other);
2. Management of privileges should not be based on identities;
3. Support of finer levels of granularity;
4. Contextual-awareness;
5. Support dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy, trust-awareness).
6. Provide dynamic policies combining access control and risk.

Prevention

To counter insider threats and minimize mismatch (inconsistency) between trust and trustworthiness following features should be supported:

1. Using of proper authentications methods (some are more secure then other);
2. Management of privileges should not be based on identities;
3. Support of finer levels of granularity;
4. Contextual-awareness;
5. Support dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy, trust-awareness).
6. Provide dynamic policies combining access control and risk.

Prevention

To counter insider threats and minimize mismatch (inconsistency) between trust and trustworthiness following features should be supported:

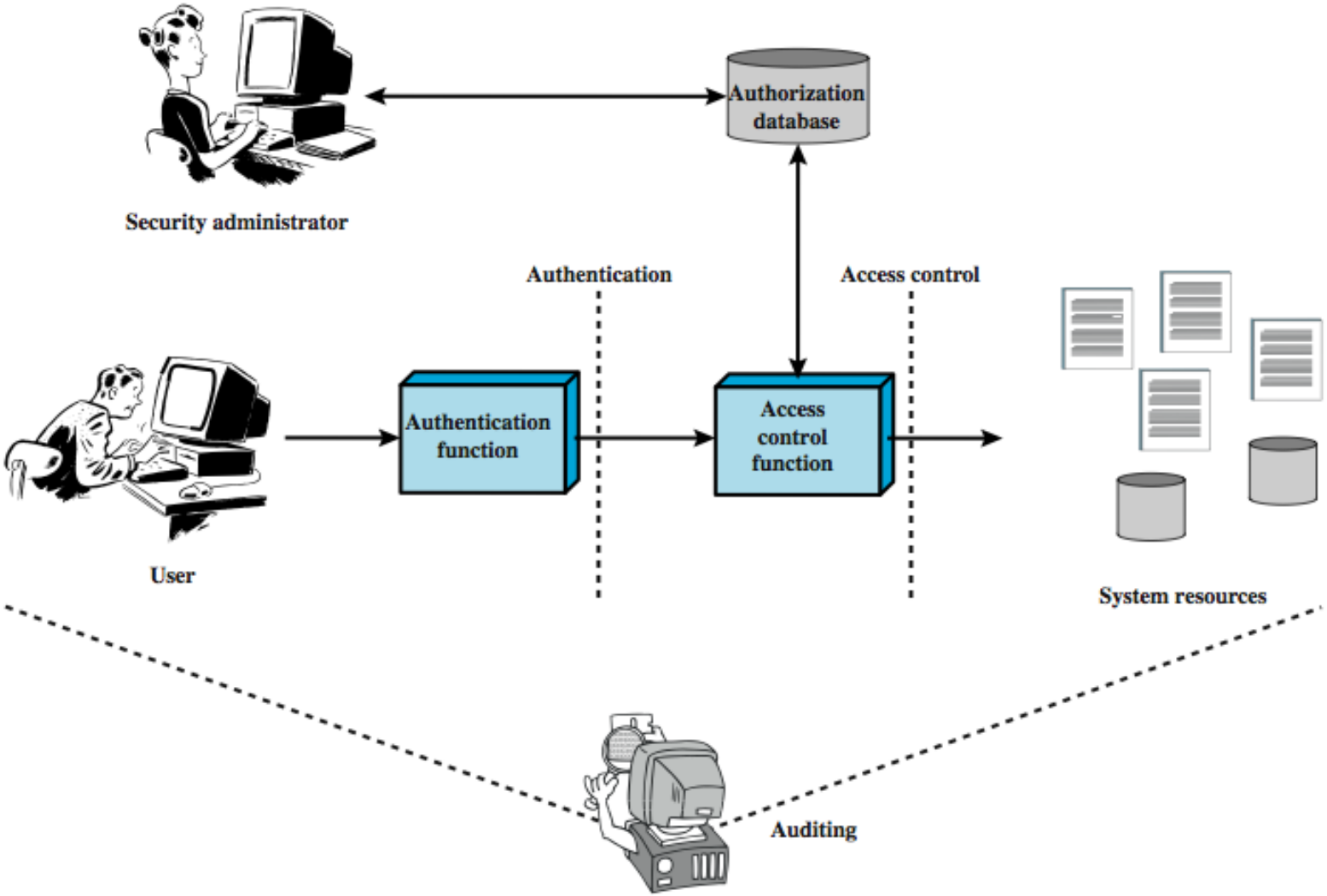
1. Using of proper authentications methods (some are more secure then other);
2. Management of privileges should not be based on identities;
3. Support of finer levels of granularity;
4. Contextual-awareness;
5. Support dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy, trust-awareness).
6. Provide dynamic policies combining access control and risk.

Prevention

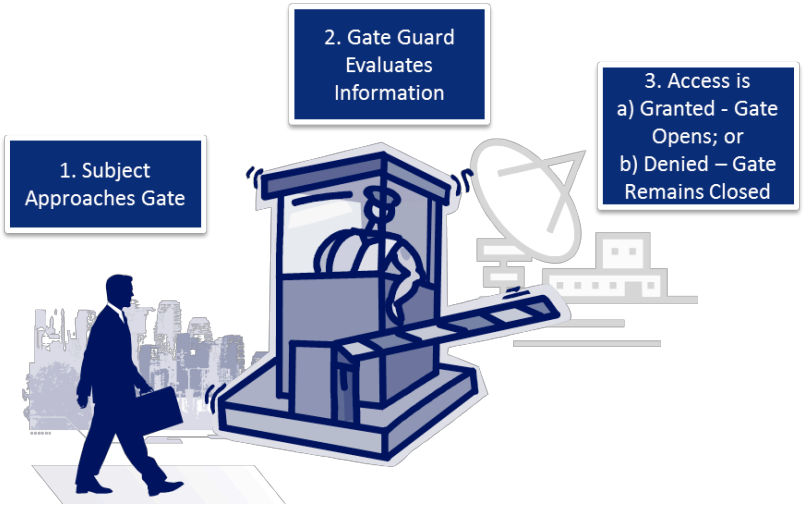
To counter insider threats and minimize mismatch (inconsistency) between trust and trustworthiness following features should be supported:

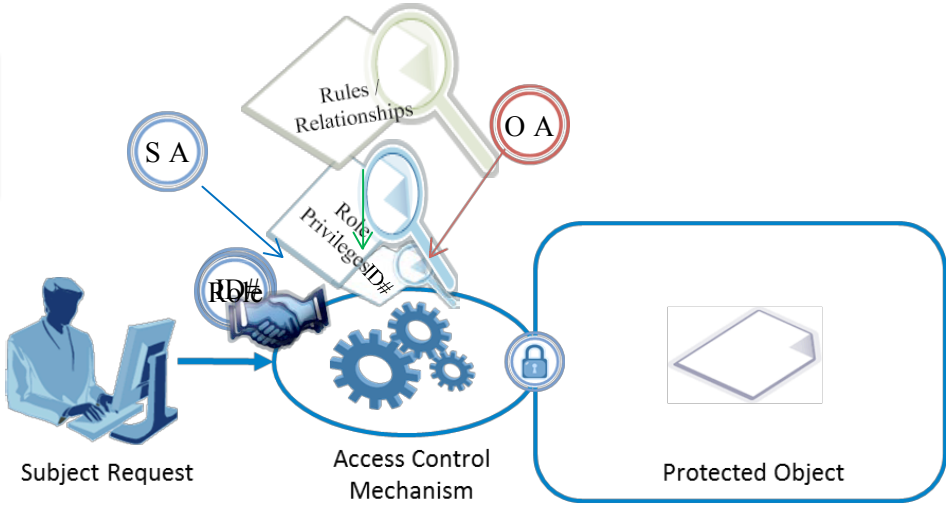
1. Using of proper authentications methods (some are more secure then other);
2. Management of privileges should not be based on identities;
3. Support of finer levels of granularity;
4. Contextual-awareness;
5. Support dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy, trust-awareness).
6. Provide dynamic policies combining access control and risk.

Access Control Model



History





MAC/DAC

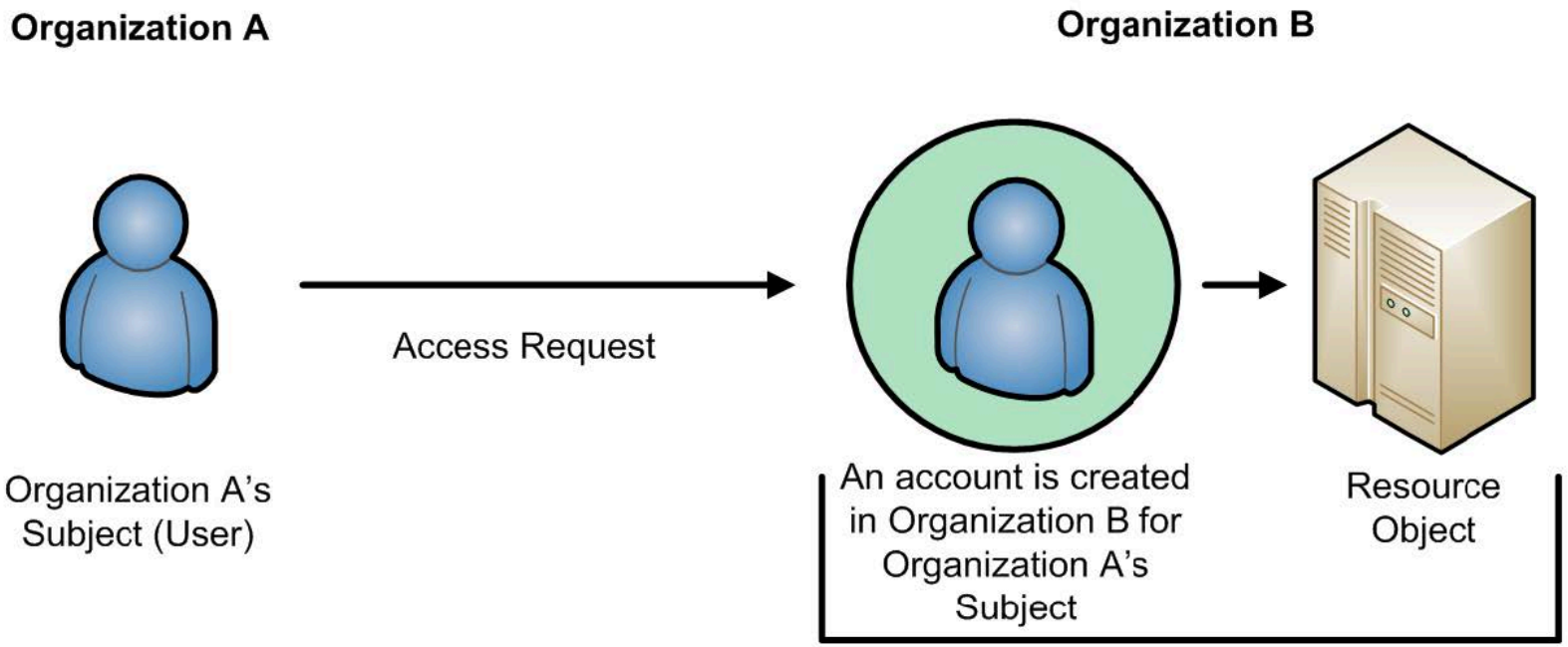
IBAC/ACL's

RBAC

ABAC

1960's 1970's 1980's 1990's 2000's Now

Multi-organizational access challenge



Organization B provisions an identity for Organization A's Subject prior to their accessing an Organization B Resource Object.

Attribute Based Access Control (ABAC)

Definition (from NIST)

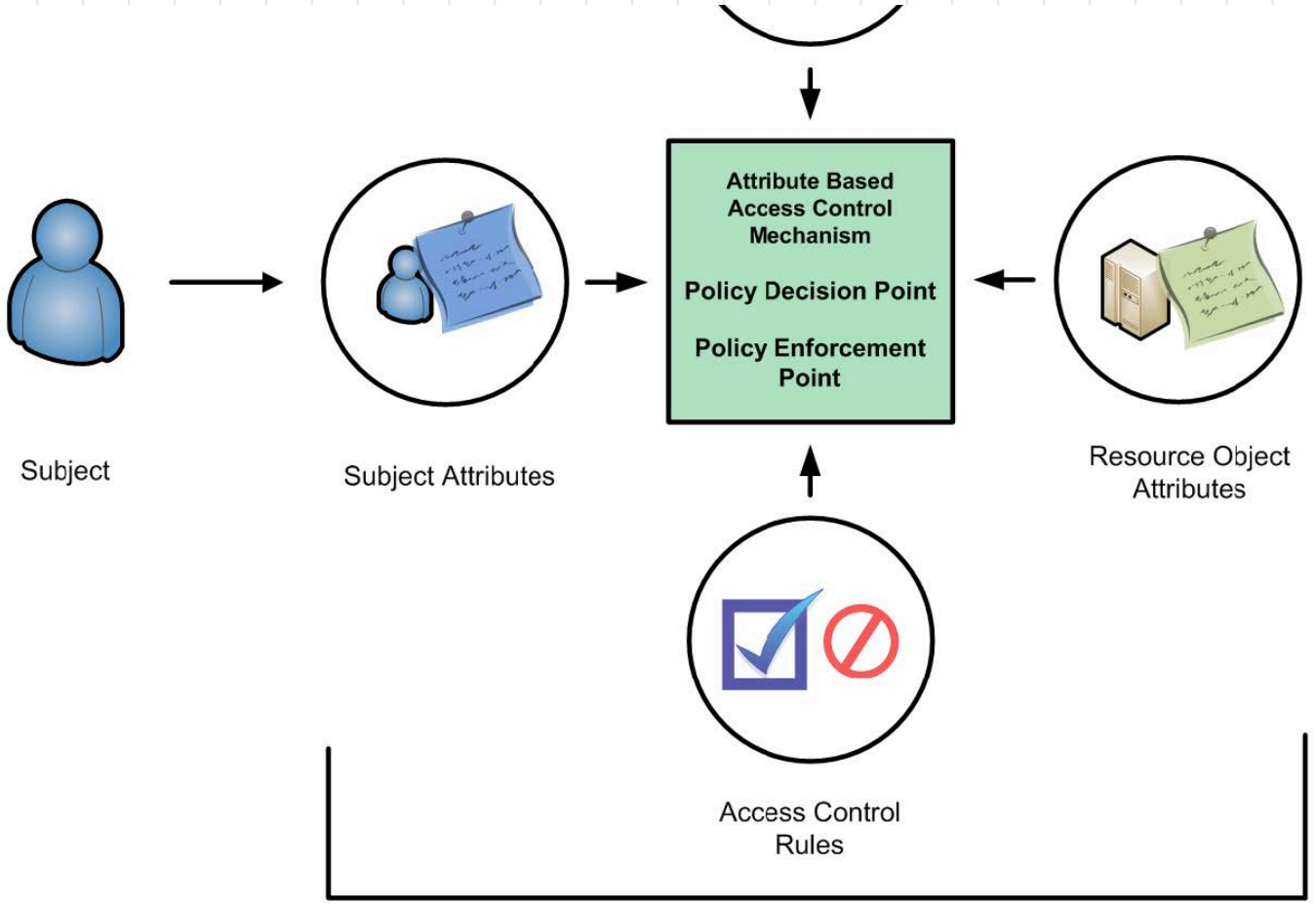
A logical access control methodology where authorization to perform a set of operations is determined by evaluating attributes associated with the subject, object, requested operations, and, in some cases, environment conditions against policy, rules, or relationships that describe the allowable operations for a given set of attributes.

Attribute Based Access Control (ABAC)

Definition (from NIST)

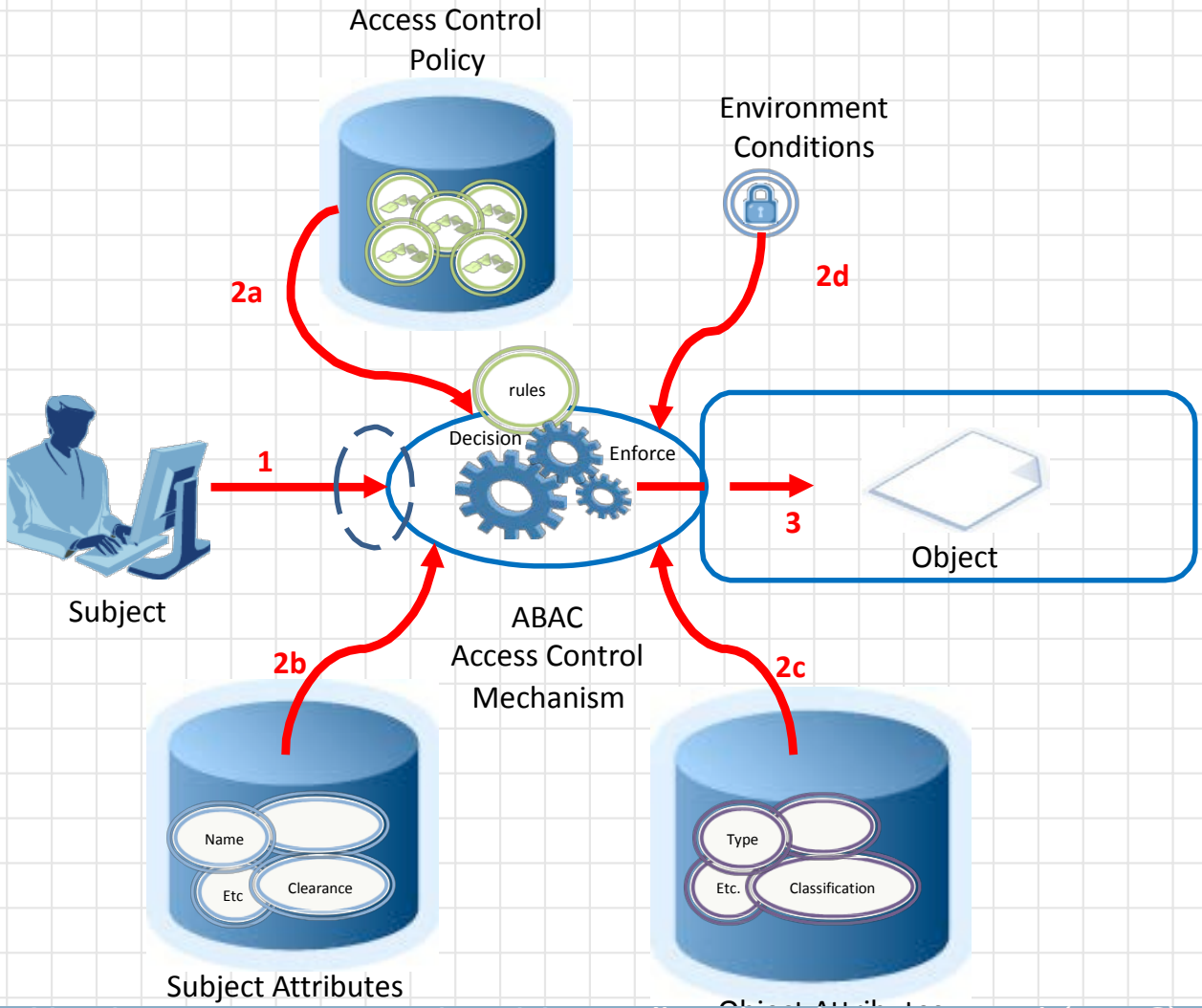
A logical access control methodology where authorization to perform a set of operations is determined by evaluating attributes associated with the subject, object, requested operations, and, in some cases, environment conditions against policy, rules, or relationships that describe the allowable operations for a given set of attributes.

Core ABAC concept

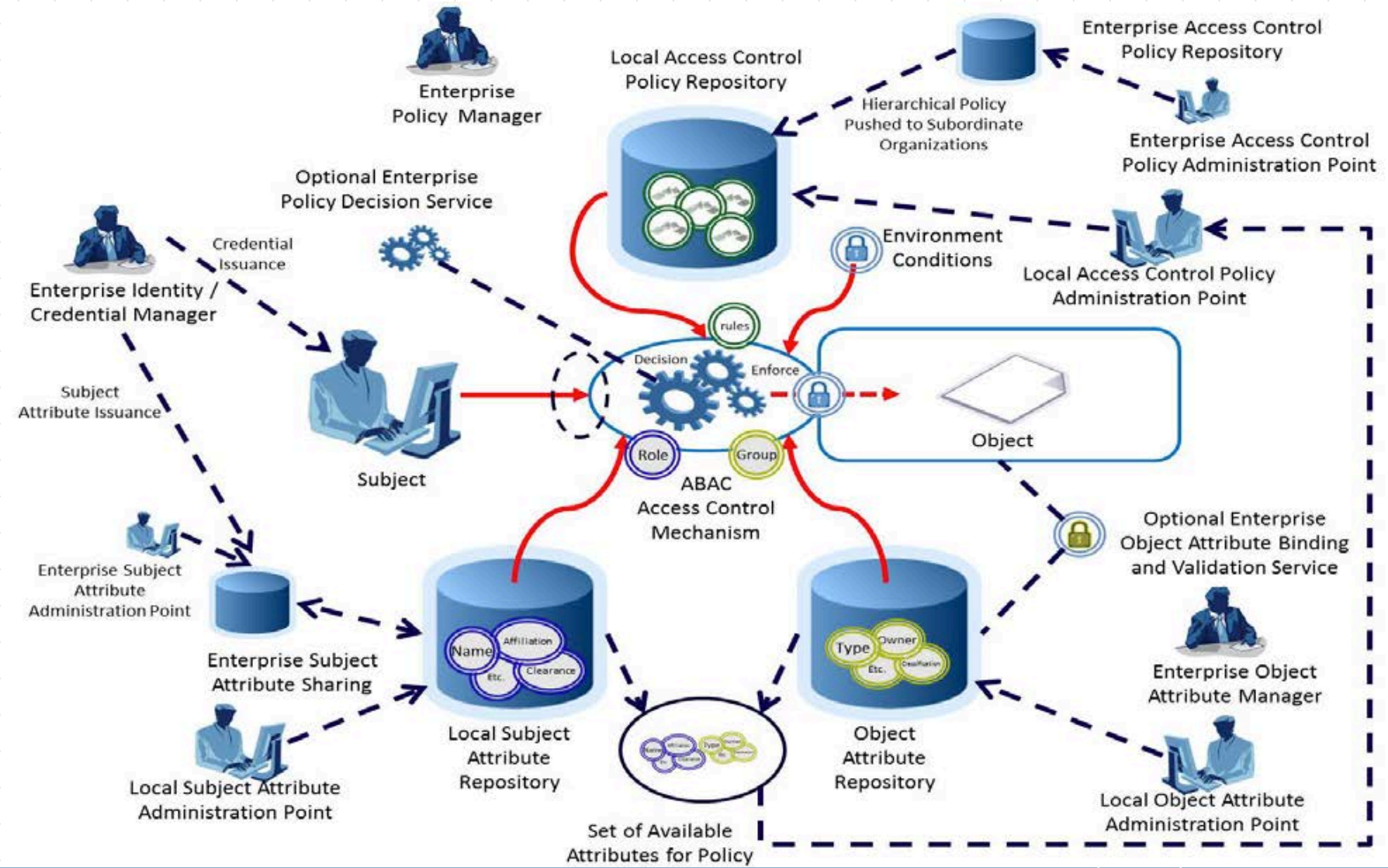


When an access request is made, Attributes and Access Control Rules are evaluated by the Attribute Based Access Control Mechanism to provide an access control decision. In ABAC's basic form, the Access Control Mechanism contains both a Policy Decision Point, and a Policy

ABAC scenario



Enterprise ABAC Scenario



ABAC summarized

Authentication and authorization are based on attributes.

- Service provider requires from a user to present a specific set of attributes certified by trusted authority (certifiers) to get access granted.
- Users get issued certified credentials containing attributes with attribute values (content) the issuer (certification authority) is vouches.

ABAC summarized

Authentication and authorization are based on attributes.

- Service provider requires from a user to present a specific set of attributes certified by trusted authority (certifiers) to get access granted.
- Users get issued certified credentials containing attributes with attribute values (content) the issuer (certification authority) is vouches.

ABAC summarized

Authentication and authorization are based on attributes.

- Service provider requires from a user to present a specific set of attributes certified by trusted authority (certifiers) to get access granted.
- Users get issued certified credentials containing attributes with attribute values (content) the issuer (certification authority) is vouches.

Advantages of ABAC

- Requires no advance knowledge of requesters.
- An individual's attributes can be correlated from multiple sources to create a unified identity.
- Highly adaptable to changing needs, efficient for agencies where individuals come and go frequently.
- Allows for fine-grained access decisions and accountability to address unique challenges like the insider threat.

Advantages of ABAC

- Requires no advance knowledge of requesters.
- An individual's attributes can be correlated from multiple sources to create a unified identity.
- Highly adaptable to changing needs; efficient for agencies where individuals come and go frequently.
- Allows for fine-grained access decisions and accountability to address unique challenges like the insider threat.

Advantages of ABAC

- Requires no advance knowledge of requesters.
- An individual's attributes can be correlated from multiple sources to create a unified identity.
- Highly adaptable to changing needs; efficient for agencies where individuals come and go frequently.
- Allows for fine-grained access decisions and accountability to address unique challenges like the insider threat.

Advantages of ABAC

- Requires no advance knowledge of requesters.
- An individual's attributes can be correlated from multiple sources to create a unified identity.
- Highly adaptable to changing needs; efficient for agencies where individuals come and go frequently.
- Allows for fine-grained access decisions and accountability to address unique challenges like the insider threat.

Advantages of ABAC

- Requires no advance knowledge of requesters.
- An individual's attributes can be correlated from multiple sources to create a unified identity.
- Highly adaptable to changing needs; efficient for agencies where individuals come and go frequently.
- Allows for fine-grained access decisions and accountability to address unique challenges like the insider threat.

Example: Constraints in ABAC

Let $\{bf_1, bf_2, \ldots bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
2. cardinality:
3. precondition:

Example: Constraints in ABAC

Let $\{bf_1, bf_2,...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

- Mutual exclusion (Separation of Duty (SoD)):
- cardinality:
- precondition:

Example: Constraints in ABAC

Let $\{bf_1, bf_2, ...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
   1. a client cannot get both benefits bf_1 and bf_2
   2. a particular employee cannot be both "programmer" and "tester" for the same project
   3. a user cannot hold attributes "president" and "vice-president"
2. cardinality:
   1. a client cannot get more then k benefits from $\{bf_1, ..., bf_n\}$, $k < n$.
3. precondition:
   1. in order to get benefit bf_3 the client first need to get benefit bf_2

Example: Constraints in ABAC

Let $\{bf_1, bf_2, ...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
   1. a client cannot get both benefits bf_1 and bf_2
   2. a particular employee cannot be both "programmer" and "tester" for the same project
   3. a user cannot hold attributes "president" and "vice-president"
2. cardinality:
   1. a client cannot get more then k benefits from $\{bf_1, ..., bf_n\}$, $k < n$.
3. precondition:
   1. in order to get benefit bf_3 the client first need to get benefit bf_2

Example: Constraints in ABAC

Let $\{bf_1, bf_2, ...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
   1. a client cannot get both benefits bf_1 and bf_2
   2. a particular employee cannot be both "programmer" and "tester" for the same project
   3. a user cannot hold attributes "president" and "vice-president"
2. cardinality:
   1. a client cannot get more then k benefits from $\{bf_1, ..., bf_n\}$, $k < n$.
3. precondition:
   1. in order to get benefit bf_3 the client first need to get benefit bf_2

Example: Constraints in ABAC

Let $\{bf_1, bf_2, ...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
   1. a client cannot get both benefits bf_1 and bf_2
   2. a particular employee cannot be both "programmer" and "tester" for the same project
   3. a user cannot hold attributes "president" and "vice-president"
2. cardinality:
   1. a client cannot get more then k benefits from $\{bf_1, ..., bf_n\}$, $k < n$.
3. precondition:
   1. in order to get benefit bf_3 the client first need to get benefit bf_2

Example: Constraints in ABAC

Let $\{bf_1, bf_2, ...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
   1. a client cannot get both benefits bf_1 and bf_2
   2. a particular employee cannot be both "programmer" and "tester" for the same project
   3. a user cannot hold attributes "president" and "vice-president"
2. cardinality:
   1. a client cannot get more then k benefits from $\{bf_1, ..., bf_n\}$, $k < n$.
3. precondition:
   1. in order to get benefit bf_i the client first need to get benefit bf_j

Example: Constraints in ABAC

Let $\{bf_1, bf_2, ...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
   1. a client cannot get both benefits bf_1 and bf_2
   2. a particular employee cannot be both "programmer" and "tester" for the same project
   3. a user cannot hold attributes "president" and "vice-president"
2. cardinality:
   1. a client cannot get more then k benefits from $\{bf_1, ..., bf_n\}$, $k < n$.
3. precondition:
   1. in order to get benefit bf_3 the client first need to get benefit bf_2

Example: Constraints in ABAC

Let $\{bf_1, bf_2, ...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
   1. a client cannot get both benefits bf_1 and bf_2
   2. a particular employee cannot be both "programmer" and "tester" for the same project
   3. a user cannot hold attributes "president" and "vice-president"
2. cardinality:
   1. a client cannot get more then k benefits from $\{bf_1, ..., bf_n\}$, $k < n$.
3. precondition:
   1. in order to get benefit bf_j the client first need to get benefit bf_i

Example: Constraints in ABAC

Let $\{bf_1, bf_2, ...bf_n\}$ be a set of benefits a customer may have (attributes or attribute values)

1. Mutual exclusion (Separation of Duty (SoD)):
   1. a client cannot get both benefits bf_1 and bf_2
   2. a particular employee cannot be both "programmer" and "tester" for the same project
   3. a user cannot hold attributes "president" and "vice-president"
2. cardinality:
   1. a client cannot get more then k benefits from $\{bf_1, ..., bf_n\}$, $k < n$.
3. precondition:
   1. in order to get benefit bf_j the client first need to get benefit bf_i

Privacy

"On the Internet, nobody knows you're a dog."

Privacy vs Security

- Privacy – the right of individuals, groups and institutions to control the collection and use of (personal) information about themselves.
- Security - protection of information from unauthorized users (preserving the integrity, availability and confidentiality (CIA) of information system resources)

Privacy vs Security

- Privacy – the right of individuals, groups and institutions to control the collection and use of (personal) information about themselves;
- Security - protection of information from unauthorized users (preserving the integrity, availability and confidentiality (CIA) of information system resources)

Privacy vs Security

- Privacy – the right of individuals, groups and institutions to control the collection and use of (personal) information about themselves;
- Security - protection of information from unauthorized users (preserving the integrity, availability and confidentiality (CIA) of information system resources)

Types of privacy

- Identity privacy: right not to reveal your identity
- Location privacy: right not to reveal your location
- Movement privacy: right not reveal your movements
- Transaction privacy: right not to reveal your transactions

Types of privacy

- Identity privacy: right not to reveal your identity
- Location privacy: right not to reveal your location
- Movement privacy: right not reveal your movements
- Transaction privacy: right not to reveal your transactions

Types of privacy

- Identity privacy: right not to reveal your identity
- Location privacy: right not to reveal your location
- Movement privacy: right not reveal your movements
- Transaction privacy: right not to reveal your transactions

Types of privacy

- Identity privacy: right not to reveal your identity
- Location privacy: right not to reveal your location
- Movement privacy: right not reveal your movements
- Transaction privacy: right not to reveal your transactions

Types of privacy

- Identity privacy: right not to reveal your identity
- Location privacy: right not to reveal your location
- Movement privacy: right not reveal your movements
- Transaction privacy: right not to reveal your transactions

Erosion of privacy

- Changes in technology (Internet of things, big data, cloud computing, cybersecurity) are making privacy harder to achieve.

Erosion of privacy

- Changes in technology (Internet of things, big data, cloud computing, cybersecurity) are making privacy harder to achieve:
  - reduced cost for data collection and storage
  - increased ability to process large amounts of data

Erosion of privacy

- Changes in technology (Internet of things, big data, cloud computing, cybersecurity) are making privacy harder to achieve:
  - reduced cost for data collection and storage
  - increased ability to process large amounts of data

Erosion of privacy

- Changes in technology (Internet of things, big data, cloud computing, cybersecurity) are making privacy harder to achieve:
  - reduced cost for data collection and storage
  - increased ability to process large amounts of data

What should be done?

- Security $\neq$ privacy.
- Radically change of reality demands new approaches to privacy
- Need means of detection, neutralization, ...
- Cryptographic protection can increase trust levels and protect privacy in some degree but usually it carries additional cost and will reduce efficiency.
- Need to find technological enforcement of privacy requirements: protect users' privacy to make new products and services possible

What should be done?

- Security $\neq$ privacy;
- Radically change of reality demands new approaches to privacy
- Need means of detection, neutralization,. . .
- Cryptographic protection can increase trust levels and protect privacy in some degree but usually it carries additional cost and will reduce efficiency.
- Need to find technological enforcement of privacy requirements: protect users' privacy to make new products and services possible

What should be done?

- Security $\neq$ privacy;
- Radically change of reality demands new approaches to privacy
- Need means of detection, neutralization,. . .
- Cryptographic protection can increase trust levels and protect privacy in some degree but usually it carries additional cost and will reduce efficiency.
- Need to find technological enforcement of privacy requirements: protect users' privacy to make new products and services possible

What should be done?

- Security $\neq$ privacy;
- Radically change of reality demands new approaches to privacy
- Need means of detection, neutralization,...
- Cryptographic protection can increase trust levels and protect privacy in some degree but usually it carries additional cost and will reduce efficiency.
- Need to find technological enforcement of privacy requirements: protect users' privacy to make new products and services possible

What should be done?

- Security $\neq$ privacy;
- Radically change of reality demands new approaches to privacy
- Need means of detection, neutralization,. . .
- Cryptographic protection can increase trust levels and protect privacy in some degree but usually it carries additional cost and will reduce efficiency.
- Need to find technological enforcement of privacy requirements: protect users' privacy to make new products and services possible

What should be done?

- Security $\neq$ privacy;
- Radically change of reality demands new approaches to privacy
- Need means of detection, neutralization,. . .
- Cryptographic protection can increase trust levels and protect privacy in some degree but usually it carries additional cost and will reduce efficiency.
- Need to find technological enforcement of privacy requirements: protect users' privacy to make new products and services possible

Trust for Privacy

Consider computer-based interaction

- Interactions (almost) always involve dissemination of private data
- Threats of privacy violations result in lower trust
- Trust must be established before a privacy disclosure
- Trustworthiness of each entities or groups of entities may be determined from context (e.g., properties of deployment area, sensor design, roles, etc.)
- Privacy and trust are closely related

Trust for Privacy

Consider computer-based interaction

- Interactions (almost) always involve dissemination of private data
- Threats of privacy violations result in lower trust
- Trust must be established before a privacy disclosure
- Trustworthiness of each entities or groups of entities may be determined from context (e.g., properties of deployment area, sensor design, roles, etc.)
- Privacy and trust are closely related

Trust for Privacy

Consider computer-based interaction

- Interactions (almost) always involve dissemination of private data
- Threats of privacy violations result in lower trust
- Trust must be established before a privacy disclosure
- Trustworthiness of each entities or groups of entities may be determined from context (e.g., properties of deployment area, sensor design, roles, etc.)
- Privacy and trust are closely related

Trust for Privacy

Consider computer-based interaction

- Interactions (almost) always involve dissemination of private data
- Threats of privacy violations result in lower trust
- Trust must be established before a privacy disclosure
- Trustworthiness of each entities or groups of entities may be determined from context (e.g., properties of deployment area, sensor design, roles, etc.)
- Privacy and trust are closely related

Trust for Privacy

Consider computer-based interaction

- Interactions (almost) always involve dissemination of private data
- Threats of privacy violations result in lower trust
- Trust must be established before a privacy disclosure
- Trustworthiness of each entities or groups of entities may be determined from context (e.g., properties of deployment area, sensor design, roles, etc.)
- Privacy and trust are closely related

Trust for Privacy

Consider computer-based interaction

- Interactions (almost) always involve dissemination of private data
- Threats of privacy violations result in lower trust
- Trust must be established before a privacy disclosure
- Trustworthiness of each entities or groups of entities may be determined from context (e.g., properties of deployment area, sensor design, roles, etc.)
- Privacy and trust are closely related

Trust for Privacy

Consider computer-based interaction

- Interactions (almost) always involve dissemination of private data
- Threats of privacy violations result in lower trust
- Trust must be established before a privacy disclosure
- Trustworthiness of each entities or groups of entities may be determined from context (e.g., properties of deployment area, sensor design, roles, etc.)
- Privacy and trust are closely related

Access control

- An access decision is based on user's identity and some secret the only user posses (something to know, something to have something to be).
- If to prove/get access to the resources the user need to reveal his attributes the privacy violation problem can arise.
- A privacy preserving attribute-based access control protects user identity and enforce access control where access is based on attributes.
- A privacy preserving trust-aware access control, enforce access to resources based on trustworthiness of users.

Access control

1. An access decision is based on user's identity and some secret the only user posses (something to know, something to have something to be).
2. If to prove/get access to the resources the user need to reveal his attributes the privacy violation problem can arise.
3. A privacy preserving attribute-based access control, protects user identity and enforce access control where access is based on attributes.
4. A privacy preserving trust-aware access control, enforce access to resources based on trustworthiness of users.

Access control

1. An access decision is based on user's identity and some secret the only user posses (something to know, something to have something to be).
2. If to prove/get access to the resources the user need to reveal his attributes the privacy violation problem can arise.
3. A privacy preserving attribute-based access control, protects user identity and enforce access control where access is based on attributes.
4. A privacy preserving trust-aware access control, enforce access to resources based on trustworthiness of users.

Access control

1. An access decision is based on user's identity and some secret the only user posses (something to know, something to have something to be).
2. If to prove/get access to the resources the user need to reveal his attributes the privacy violation problem can arise.
3. A privacy preserving attribute-based access control, protects user identity and enforce access control where access is based on attributes.
4. A privacy preserving trust-aware access control, enforce access to resources based on trustworthiness of users.

Access control

1. An access decision is based on user's identity and some secret the only user posses (something to know, something to have something to be).
2. If to prove/get access to the resources the user need to reveal his attributes the privacy violation problem can arise.
3. A privacy preserving attribute-based access control, protects user identity and enforce access control where access is based on attributes.
4. A privacy preserving trust-aware access control, enforce access to resources based on trustworthiness of users.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:
  1. Online: SAML, OpenID etc
     - Privacy violation in both cases
  2. Off-line: X.509 certificates
     - Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:
  1. Online: SAML, OpenID etc
     - Privacy violation in both cases
  2. Off-line: X.509 certificates
     - Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:

1. Online: SAML, OpenID etc
   - creats tokens on demand and may track users' on-line activities
   - able to impersonate their users.
2. Off-line: X.509 certificates
   - can link user's online transactions over different domains since they force users to reveal more attributes than actually needed.

- Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:

1. Online: SAML, OpenID etc
   - create tokens on demand and may track users' on-line activities
   - able to impersonate their users.
2. Off-line: X.509 certificates
   - can link user's online transactions over different domains since they force users to reveal more attributes than actually needed.

- Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:

1. Online: SAML, OpenID etc
   - create tokens on demand and may track users' on-line activities
   - able to impersonate their users.
2. Off-line: X.509 certificates
   - can link user's on-line transactions over different domains since they force users to reveal more attributes than actually needed.

- Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:

1. Online: SAML, OpenID etc
   - create tokens on demand and may track users' on-line activities
   - able to impersonate their users.
2. Off-line: X.509 certificates
   - can link user's on-line transactions over different domains since they force users to reveal more attributes than actually needed.

- Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:

1. Online: SAML, OpenID etc
   - create tokens on demand and may track users' on-line activities
   - able to impersonate their users.
2. Off-line: X.509 certificates
   - can link user's on-line transactions over different domains since they force users to reveal more attributes than actually needed.

- Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:

1. Online: SAML, OpenID etc
   - create tokens on demand and may track users' on-line activities
   - able to impersonate their users.
2. Off-line: X.509 certificates
   - can link user's on-line transactions over different domains since they force users to reveal more attributes than actually needed.

- Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:

1. Online: SAML, OpenID etc
   - create tokens on demand and may track users' on-line activities
   - able to impersonate their users.
2. Off-line: X.509 certificates
   - can link user's on-line transactions over different domains since they force users to reveal more attributes than actually needed.

- Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Privacy treats

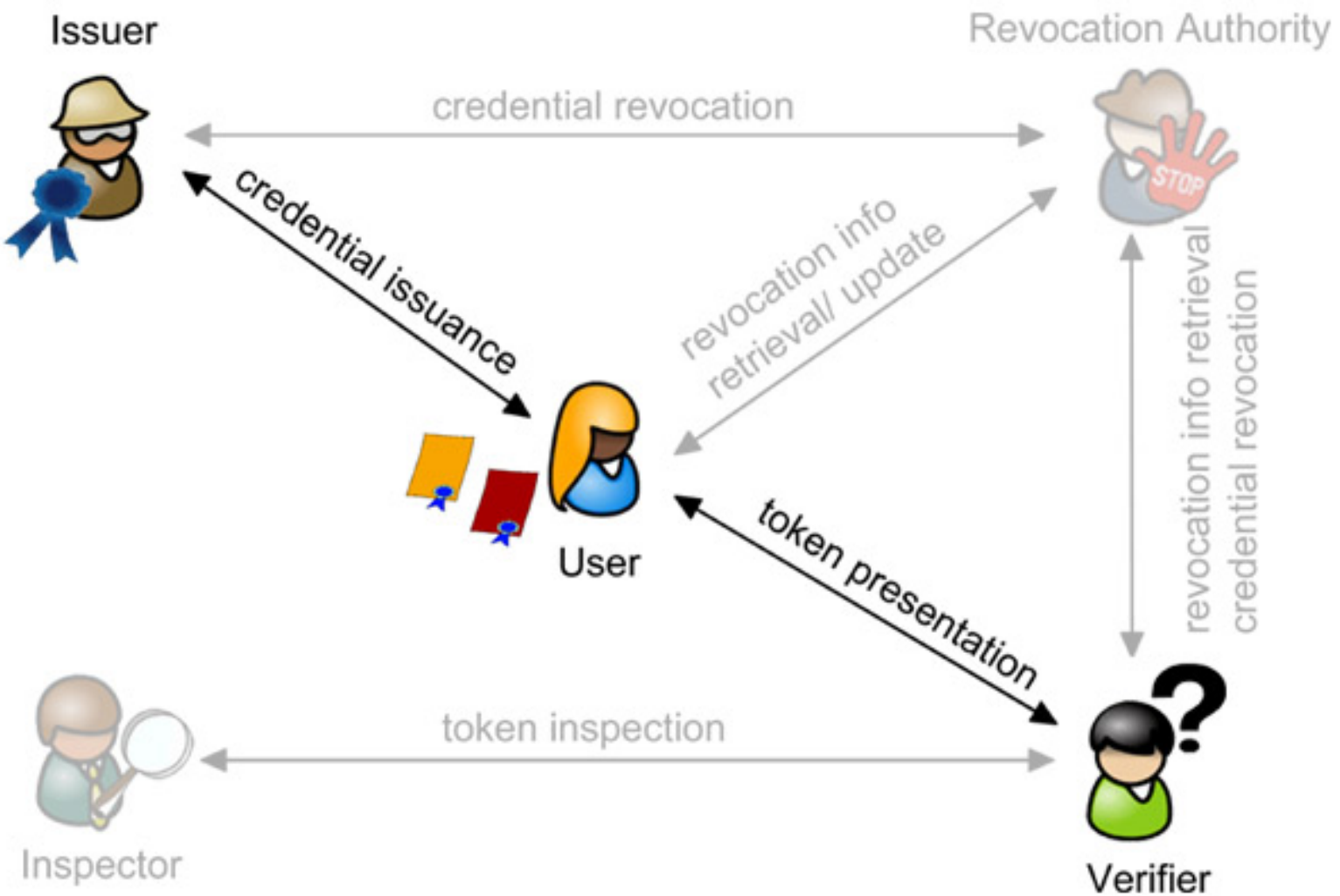
To use attributes in access control they have to be trusted - certified by trusted certification authority.

- Two main approaches:

1. Online: SAML, OpenID etc
   - create tokens on demand and may track users' on-line activities
   - able to impersonate their users.
2. Off-line: X.509 certificates
   - can link user's on-line transactions over different domains since they force users to reveal more attributes than actually needed.

- Privacy violation in both cases
- Solution: In privacy-preserving authentication schemes users derive unlinkable tokens offline from certified attributes they have preliminary received from trusted certification authorities.

Entities


Issuer
Revocation Authority
credential revocation
credential issuance
revocation info
retrieval/ update
revocation info retrieval
credential revocation
User
token presentation
token inspection
Inspector
Verifier

Cryptographic Realization: Blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m: and gets blinded message $\bar{m}$: $\bar{m} = b(m)$
- Signer: signs $\bar{m}$ and produces blinded signature $\bar{\sigma}$ of m
- Originator: unblinds $\bar{\sigma}$ and gets valid signature σ of m: $\sigma = b^{-1}(\bar{\sigma})$

Cryptographic Realization: Blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\bar{m}$: $\bar{m} = b(m)$
- Signer: signs $\bar{m}$ and produces blinded signature $\bar{\sigma}$ of m
- Originator: unblinds $\bar{\sigma}$ and gets valid signature σ of m: $\sigma = b^{-1}(\bar{\sigma})$

Cryptographic Realization: Blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m without the signer being able to learn anything about m at the time of signing.

Formally:

- Originator: blinds m, and gets blinded message $\bar{m}$: $\bar{m} = b(m)$
- Signer: signs $\bar{m}$ and produces blinded signature $\bar{\sigma}$ of m
- Originator: unblinds $\bar{\sigma}$ and gets valid signature σ of m: $\sigma = b^{-1}(\bar{\sigma})$

Cryptographic Realization: Blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\overline{m}$: $\overline{m} = B(m)$
- Signer: signs $\overline{m}$ and produces blinded signature $\overline{\sigma}$ of m
- Originator: unblinds $\overline{\sigma}$ and gets valid signature σ of m: $\sigma = B^{-1}(\overline{\sigma})$

Cryptographic Realization: Blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\overline{m}$: $\overline{m} = B(m)$
- Signer: signs $\overline{m}$ and produces blinded signature $\overline{\sigma}$ of m
- Originator: unblinds $\overline{\sigma}$ and gets valid signature σ of m: $\sigma = B^{-1}(\overline{\sigma})$

Cryptographic Realization: Blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\overline{m}$: $\overline{m} = B(m)$
- Signer: signs $\overline{m}$ and produces blinded signature $\overline{\sigma}$ of m
- Originator: unblinds $\overline{\sigma}$ and gets valid signature σ of m: $\sigma = B^{-1}(\overline{\sigma})$

Cryptographic Realization: Partially blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m and some common information without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\overline{m}$: $\overline{m} = b(m)$
- Signer: signs $\overline{m}$, info and produces partially blinded signature $\overline{\sigma}$ of m, info
- Originator: unblinds $\overline{\sigma}$ and gets valid signature σ of m, info: $\sigma = b^{-1}(\overline{\sigma})$

Cryptographic Realization: Partially blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m and some common information without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\overline{m}$: $\overline{m} = b(m)$
- Signer: signs $\overline{m}$, info and produces partially blinded signature $\overline{\sigma}$ of $\overline{m}$, info
- Originator: unblinds $\overline{\sigma}$ and gets valid signature σ of m, info: $\sigma = b^{-1}(\overline{\sigma})$

Cryptographic Realization: Partially blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m and some common information without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\overline{m}$: $\overline{m} = B(m)$
- Signer: signs $\overline{m}, info$ and produces partially blinded signature $\overline{\sigma}$ of $m, info$
- Originator: unblinds $\overline{\sigma}$ and gets valid signature σ of $m, info$: $\sigma = B^{-1}(\overline{\sigma})$

Cryptographic Realization: Partially blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m and some common information without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\overline{m}$: $\overline{m} = B(m)$
- Signer: signs $\overline{m}, info$ and produces partially blinded signature $\overline{\sigma}$ of $m, info$
- Originator: unblinds $\overline{\sigma}$ and gets valid signature σ of $m, info$: $\sigma = B^{-1}(\overline{\sigma})$

Cryptographic Realization: Partially blind signature

Cryptographic primitive that provides the following functionality:
An entity (originator) is able to obtain a valid signature σ on a message m and some common information without the signer being able to learn anything about m at the time of signing.
Formally:

- Originator: blinds m, and gets blinded message $\overline{m}$: $\overline{m} = B(m)$
- Signer: signs $\overline{m}, info$ and produces partially blinded signature $\overline{\sigma}$ of $m, info$
- Originator: unblinds $\overline{\sigma}$ and gets valid signature σ of $m, info$: $\sigma = B^{-1}(\overline{\sigma})$

Privacy preserving SoD validation

Verifier creates anonymous unlinkable one-show token:

- User has an initial signed token from Verifier (t, σ)
- User creates a new token t' and sends $[(t, \sigma), B(t')]$ to Verifier
- If signature σ is valid, Verifier produces partially blind signature σ of (t', ID), where ID is a unique name identifying f ex list of conflicting attributes
- User unblinds $\bar{\sigma}$ and gets valid signature σ of (t', ID)
- t' is unlinkable to t, must be used only with attributes from ID list and after using the first time will be added to Black list by Verifier (to provide one-time showness)

Privacy preserving SoD validation

Verifier creates anonymous unlinkable one-show token:

- User has an initial signed token from Verifier (t, σ)
- User creates a new token t' and sends $[(t, \sigma), B(t')]$ to Verifier
- If signature σ is valid, Verifier produces partially blind signature $\overline{\sigma}$ of (t', ID), where ID is a unique name identifying f ex list of conflicting attributes
- User unblinds $\overline{\sigma}$ and gets valid signature σ of (t', ID)
- t' is unlinkable to t, must be used only with attributes from ID list and after using the first time will be added to Black list by Verifier (to provide one-time showness).

Privacy preserving SoD validation

Verifier creates anonymous unlinkable one-show token:

- User has an initial signed token from Verifier (t, σ)
- User creates a new token t' and sends $[(t, \sigma), B(t')]$ to Verifier
- If signature σ is valid, Verifier produces partially blind signature $\overline{\sigma}$ of (t', ID), where ID is a unique name identifying f ex list of conflicting attributes
- User unblinds $\overline{\sigma}$ and gets valid signature σ of (t', ID)
- t' is unlinkable to t, must be used only with attributes from ID list and after using the first time will be added to Black list by Verifier (to provide one-time showness).

Privacy preserving SoD validation

Verifier creates anonymous unlinkable one-show token:

- User has an initial signed token from Verifier (t, σ)
- User creates a new token t' and sends $[(t, \sigma), B(t')]$ to Verifier
- If signature σ is valid, Verifier produces partially blind signature $\overline{\sigma}$ of (t', ID), where ID is a unique name identifying f ex list of conflicting attributes
- User unblinds $\overline{\sigma}$ and gets valid signature σ of (t', ID)
- t' is unlinkable to t, must be used only with attributes from ID list and after using the first time will be added to Black list by Verifier (to provide one-time showness).

Privacy preserving SoD validation

Verifier creates anonymous unlinkable one-show token:

- User has an initial signed token from Verifier (t, σ)
- User creates a new token t' and sends $[(t, \sigma), B(t')]$ to Verifier
- If signature σ is valid, Verifier produces partially blind signature $\overline{\sigma}$ of (t', ID), where ID is a unique name identifying f ex list of conflicting attributes
- User unblinds $\overline{\sigma}$ and gets valid signature σ of (t', ID)
- t' is unlinkable to t, must be used only with attributes from ID list and after using the first time will be added to Black list by Verifier (to provide one-time showness).

Privacy preserving SoD validation

Verifier creates anonymous unlinkable one-show token:

- User has an initial signed token from Verifier (t, σ)
- User creates a new token t' and sends $[(t, \sigma), B(t')]$ to Verifier
- If signature σ is valid, Verifier produces partially blind signature $\overline{\sigma}$ of (t', ID), where ID is a unique name identifying f ex list of conflicting attributes
- User unblinds $\overline{\sigma}$ and gets valid signature σ of (t', ID)
- t' is unlinkable to t, must be used only with attributes from ID list and after using the first time will be added to Black list by Verifier (to provide one-time showness).

Trust: where and how?

- Assign levels of trustworthiness to users
- Assign levels of trustworthiness to attributes
- Define trustworthiness constraints
- Subject must satisfy trustworthiness constraints to get request granted (in addition to get it granted according to policy)

Trust: where and how?

- Assign levels of trustworthiness to users
- Assign levels of trustworthines to attributes
- Define trustworthiness constraints
- Subject must satisfy trustworthiness constraints to get request granted (in addition to get it granted according to policy)

Trust: where and how?

- Assign levels of trustworthiness to users
- Assign levels of trustworthines to attributes
- Define trustworthiness constraints
- Subject must satisfy trustworthiness constraints to get request granted (in addition to get it granted according to policy)

Trust: where and how?

- Assign levels of trustworthiness to users
- Assign levels of trustworthines to attributes
- Define trustworthiness constraints
- Subject must satisfy trustworthiness constraints to get request granted (in addition to get it granted according to policy)

Trust: where and how?

- Assign levels of trustworthiness to users
- Assign levels of trustworthines to attributes
- Define trustworthiness constraints
- Subject must satisfy trustworthiness constraints to get request granted (in addition to get it granted according to policy)

Measuring trust: subjective logic

- An opinion about trustworthiness of A:

$$\omega^A = \{t^A, d^A, u^A\}, \; t^A + d^A + u^A = 1$$

where t^A, d^A, u^A denote trust, distrust and uncertainty.

- Subjective logic defines operators to combine opinions such as Conjunction, Recommendation, Consensus, etc.

Measuring trust: subjective logic

- An opinion about trustworthiness of A:

$$\omega^A = \{t^A, d^A, u^A\},\ t^A + d^A + u^A = 1$$

where t^A, d^A, u^A denote trust, distrust and uncertainty.

- Subjective logic defines operators to combine opinions such as Conjunction, Recommendation, Consensus, etc.

Measuring trust: subjective logic

- An opinion about trustworthiness of A:

$$\omega^A = \{t^A, d^A, u^A\},\ t^A + d^A + u^A = 1$$

  where t^A, d^A, u^A denote trust, distrust and uncertainty.
- Subjective logic defines operators to combine opinions such as Conjunction, Recommendation, Consensus, etc.

Conjunction of opinions

$$\omega^s_{a_i} = \left\{t^s_{a_i}, d^s_{a_i}, u^s_{a_i}\right\}, i = 1, 2$$
$$\omega^s_{a_1 \wedge a_2} = \omega^s_{a_1} \wedge \omega^s_{a_2} = \left\{t^s_{a_1 \wedge a_2}, d^s_{a_1 \wedge a_2}, u^s_{a_1 \wedge a_2}\right\}$$
$$t^s_{a_1 \wedge a_2} = t^s_{a_1} t^s_{a_2}$$
$$d^s_{a_1 \wedge a_2} = d^s_{a_1} + d^s_{a_2} - d^s_{a_1} d^s_{a_2}$$
$$u^s_{a_1 \wedge a_2} = t^s_{a_1} u^s_{a_2} + u^s_{a_1} t^s_{a_2} + u^s_{a_1} u^s_{a_2}$$

Recommendation operator

When A does not have direct opinion ω_p^A about p, A needs to deduce indirect opinion ω_p^{AB} about trustworthiness of p based on recommendation of B and opinion ω_B^A of A about trustworthiness of recommendation of B:

$$\omega_p^{AB} = \omega_B^A \otimes \omega_p^B = \{t_p^{AB}, d_p^{AB}, u_p^{AB}\} \quad where$$
$$t_p^{AB} = t_B^A t_p^B;$$
$$d_p^{AB} = t_B^A d_p^B;$$
$$u_p^{AB} = d_B^A + u_B^A + t_B^A u_p^B$$

Consensus of opinions

Two independent opinions ω^A and ω^B about the same event can be combined into new opinion ω by consensus operator $\oplus$:

$$
\begin{aligned}
&\omega = \omega^A \oplus \omega^B \quad where \\
&t = (t^A u^B + t^B u^A)/(u^A + u^B - u^A u^B) \\
&d = \left(d^A u^B + d^B u^A\right) / \left(u^A + u^B - u^A u^B\right) \\
&u = \left(u^A u^B\right) / \left(u^A + u^B - u^A u^B\right)
\end{aligned}
$$

Attribute delegation: informal example

Objective: Support delegation of $attr$ by u to u' can be seen as recommendation of u to the AC system to accept $attr$ from u' with trustworthiness:

$$\hat{\omega}_{attr} = \omega_u \otimes \omega_{attr}$$

where
ω_u denotes trustworthiness of u
$\hat{\omega}_{attr}$ denotes trustworthiness of $attr$ delegated to u' by u
ω_{attr} denotes trustworthiness of $attr$ issued to u

Attribute delegation: informal example

Objective: Support delegation of $attr$ by u to u' can be seen as recommendation of u to the AC system to accept $attr$ from u' with trustworthiness:

$$\hat{\omega}_{attr} = \omega_u \otimes \omega_{attr}$$

where

ω_u denotes trustworthiness of u

$\hat{\omega}_{attr}$ denotes trustworthiness of $attr$ delegated to u' by u

ω_{attr} denotes trustworthiness of $attr$ issued to u

Attribute delegation: informal example

Objective: Support delegation of $\mathfrak{attr}$ by $\mathfrak{u}$ to $\mathfrak{u}'$ can be seen as recommendation of $\mathfrak{u}$ to the AC system to accept $\mathfrak{attr}$ from $\mathfrak{u}'$ with trustworthiness:

$$\hat{\omega}_{\mathfrak{attr}} = \omega_{\mathfrak{u}} \otimes \omega_{\mathfrak{attr}}$$

where

$\omega_{\mathfrak{u}}$ denotes trustworthiness of $\mathfrak{u}$

$\hat{\omega}_{\mathfrak{attr}}$ denotes trustworthiness of $\mathfrak{attr}$ delegated to $\mathfrak{u}'$ by $\mathfrak{u}$

$\omega_{\mathfrak{attr}}$ denotes trustworthiness of $\mathfrak{attr}$ issued to $\mathfrak{u}$

Attribute delegation: informal example

Objective: Support delegation of attr by $\mathfrak{u}$ to $\mathfrak{u}'$ can be seen as recommendation of $\mathfrak{u}$ to the AC system to accept attr from $\mathfrak{u}'$ with trustworthiness:

$$\hat{\omega}_{\mathrm{attr}} = \omega_{\mathfrak{u}} \otimes \omega_{\mathrm{attr}}$$

where
$\omega_{\mathfrak{u}}$ denotes trustworthiness of $\mathfrak{u}$
$\hat{\omega}_{\mathrm{attr}}$ denotes trustworthiness of attr delegated to $\mathfrak{u}'$ by $\mathfrak{u}$
ω_{attr} denotes trustworthiness of attr issued to $\mathfrak{u}$

Dynamic policies: informal example

Objective: Support for dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy).

- Trustworthiness of each user will be tracked and modified with respect of his activity in the system.
- All requests will be evaluated with respect of current user's trustworthiness level - in case it is under threshold, the request will be denied.

Dynamic policies: informal example

Objective: Support for dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy).

- Trustworthiness of each user will be tracked and modified with respect of his activity in the system.
- All requests will be evaluated with respect of current user's trustworthiness level - in case it is under threshold, the request will be denied.

Dynamic policies: informal example

Objective: Support for dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy).

- Trustworthiness of each user will be tracked and modified with respect of his activity in the system.
- All requests will be evaluated with respect of current user's trustworthiness level - in case it is under threshold, the request will be denied.

Dynamic policies: informal example

Objective: Support for dynamic policies (privileges user authorized for can be modified automatically when a user is suspected to be untrustworthy).

- Trustworthiness of each user will be tracked and modified with respect of his activity in the system.
- All requests will be evaluated with respect of current user's trustworthiness level - in case it is under threshold, the request will be denied.

Conclusion

Andrew Grove, co-founder and former CEO of Intel Corporation ("What I've Learned: Andy Grove", Esquire magazine, May 1, 2000):

> Privacy is one of the biggest problems in this new electronic age. At the heart of the Internet culture is a force that wants to find out everything about you. And once it has found out everything about you and two hundred million others, that's a very valuable asset, and people will be tempted to trade and do commerce with that asset. This wasn't the information that people were thinking of when they called this the information age.

Conclusion

Andrew Grove, co-founder and former CEO of Intel Corporation ("What I've Learned: Andy Grove", Esquire magazine, May 1, 2000):

> Privacy is one of the biggest problems in this new electronic age. At the heart of the Internet culture is a force that wants to find out everything about you. And once it has found out everything about you and two hundred million others, that's a very valuable asset, and people will be tempted to trade and do commerce with that asset. This wasn't the information that people were thinking of when they called this the information age.

Questions?

Thank you!

vladimir.oleshchuk@uia.no