Continuous Security Policy Enforcement in Streaming Environments

Rimma V. Nehme, Hyo-Sang Lim, Elisa Bertino
CERIAS, Purdue University
rnehme@cs.purdue.edu, hslim@cs.purdue.edu, bertino@cs.purdue.edu

Motivating Example: Patient Monitoring

Data Providers (e.g., patients): security preferences are continuously changing according to the data provider's policy
- Prevent insurance company to access any information
- Allow only my doctor to access my heart beat rate
- Prevent my company to access my data

Continuous Security Policy Enforcement: Overview

The Goal: Enforce access control in dynamic streaming environments where
1) data provider's security preferences may continuously change and
2) query specifier's access privileges may continuously change

Proposed Solution:
- Security Punctuations: streaming security meta-data tuples describing both data and queries' security restrictions
- Continuous Policy Queries: predefined logics for generating proper access privileges of query specifiers based on the context data streams

System Architecture

Data Providers (e.g., patients): data side security policy dynamicity
- DSMS: DTSGMS
- DSMS: DSMS
- DSMS: DSMS
- DSMS: DSMS

Continuous Security Policy Enforcement: Overview

The Goal: Enforce access control in dynamic streaming environments where
1) data provider's security preferences may continuously change and
2) query specifier's access privileges may continuously change

Proposed Solution:
- Security Punctuations: streaming security meta-data tuples describing both data and queries' security restrictions
- Continuous Policy Queries: predefined logics for generating proper access privileges of query specifiers based on the context data streams

Security-Aware Query Processing: An Example

Continuous Security Policy Queries (CP-Queries)

CREATE CONTINUOUS POLICY QUERY name AS
  --INPUT STREAM
  SELECT select_clause
  FROM from_clause
  WHERE where_clause
  --OUTPUT STREAM
  CASE expression
    WHEN value1 THEN RETURN query_security_punctuation1
    WHEN value2 THEN RETURN query_security_punctuation2
    WHEN value3 THEN return query_security_punctuation3
    [ELSE query_security_punctuation3]
  END

- Input: context data streams
- Output: meta streams, i.e., streams composed of query security punctuations (qsp)
- Processing: similar to traditional continuous queries
- QSPs are produced incrementally by CP-Query

Contributions:
- A mechanism for enforcing dynamic access control on streaming data
- Support both data-and-query-side dynamicity of access control policies
- Proposed a symmetric model to describe data-and-query side dynamicity