2008/10/8 CSS2008

データ工学とセキュリティ

東京工業大学
横田　治夫
yokota@cs.titech.ac.jp

Agenda

- データ工学とその動向
- 関係データベースとセキュリティ
- データマイニングとセキュリティ
- XMLデータベースとセキュリティ
- ストレージとセキュリティ
- ネットワークストレージとセキュリティ
  - 我々の取り組み
- まとめ

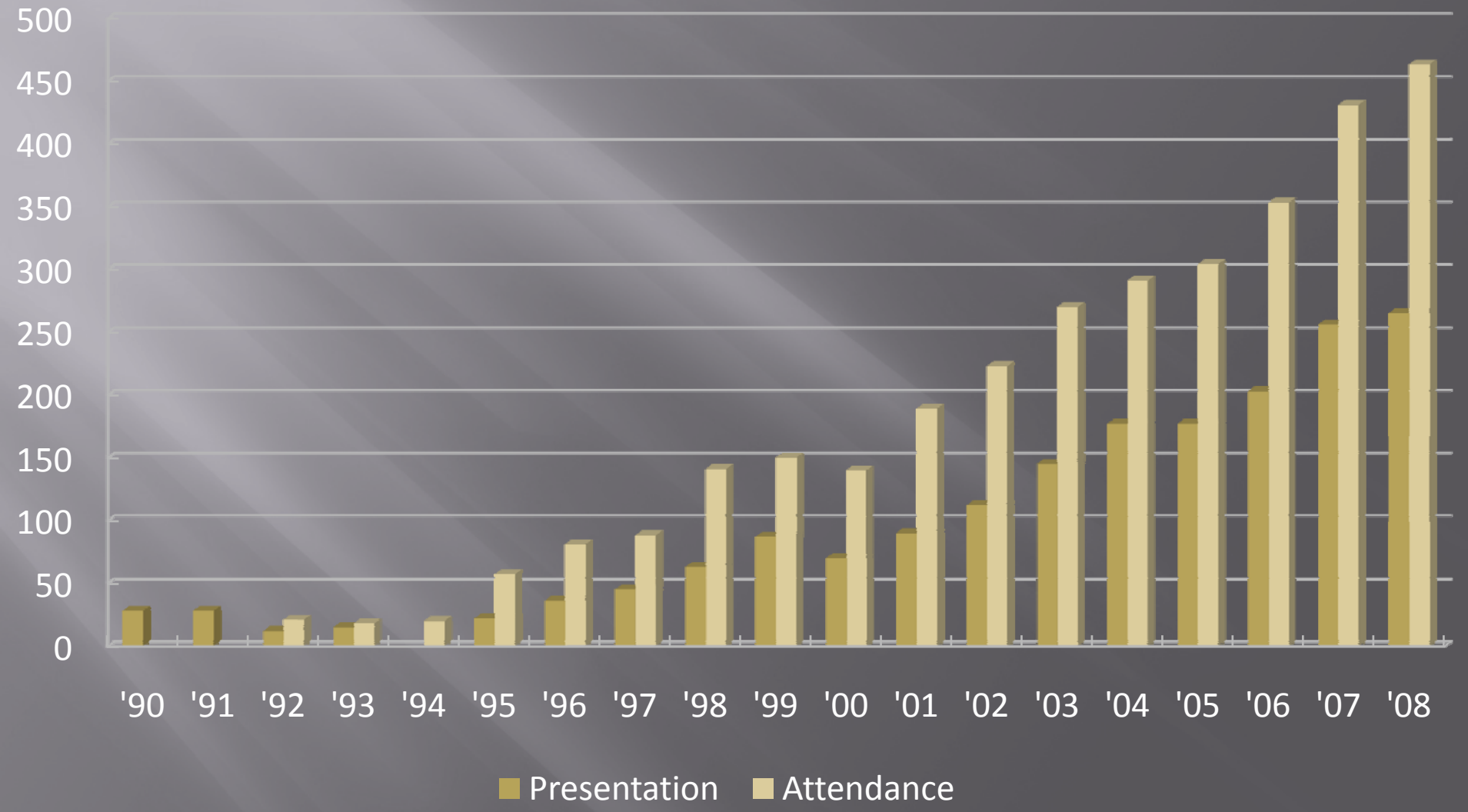
データ工学

- データを効率よく蓄積・処理して、有効に活用
  - データベース
  - 情報検索
  - ファイルシステム
  - ストレージ
  - …
- データベースを含む
  - 1980年代中頃から
    - データベースは1960年代から
    - 関係データベースの提案は1970年
- ディペンダビリティ・セキュリティはますます重要に

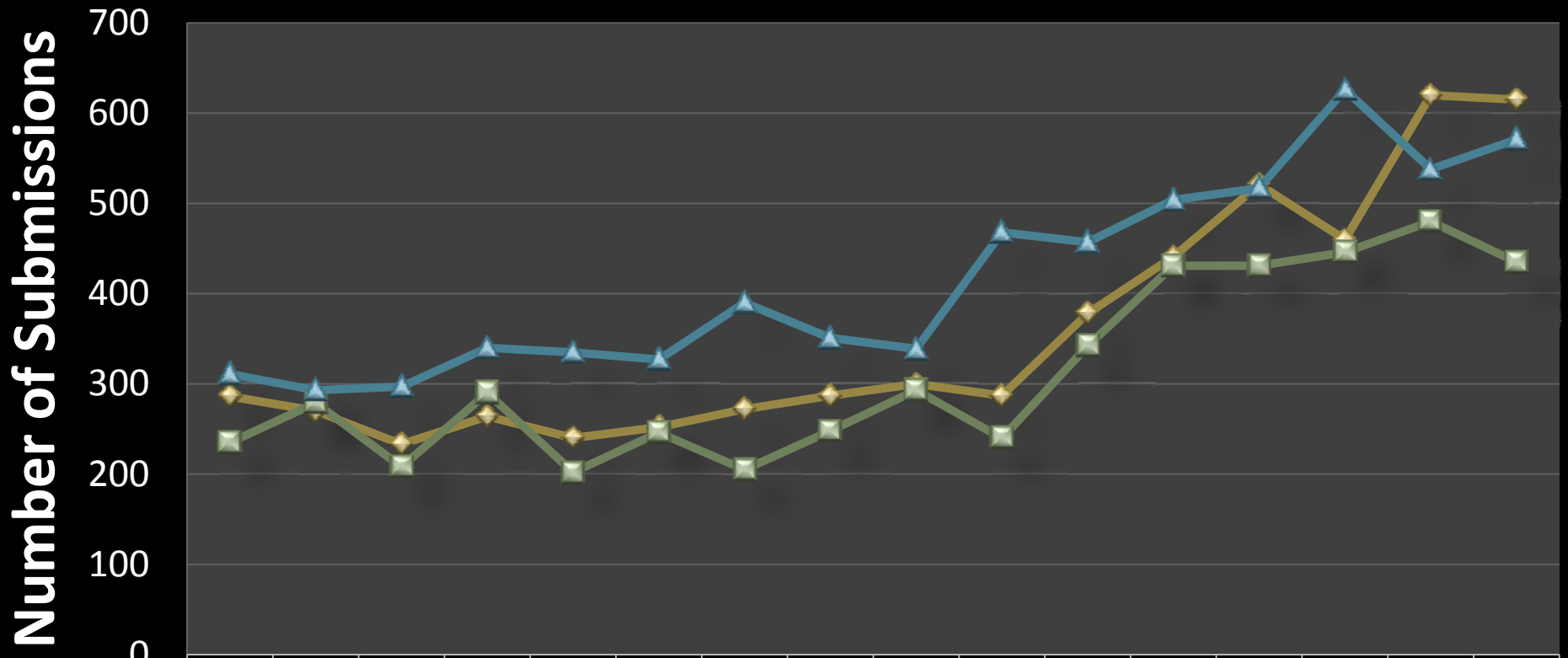
データ工学関連学会

- 電子情報通信学会データ工学研究専門委員会
  - 1986年に設立
    - （2003年、2004年委員長）
  - 毎年3月に国内WS: DEWS (Data Engineering Workshop)
    - 2008年3月で19回目
  - 他国内関係学会
    - 情報処理学会データベースシステム研究会
    - 日本データベース学会
- ICDE: International Conference on Data Engineering
  - IEEE の国際会議 （DB関係の3トップ国際会議の一つ）
    - トップ3国際会議: SIGMOD, VLDB, ICDE
  - 1984年に第1回開催（LA）
    - 2005年（第21回）東京開催
    - 2008年（第24回）はカンクン開催

成長するデータ工学（DEWSの履歴）

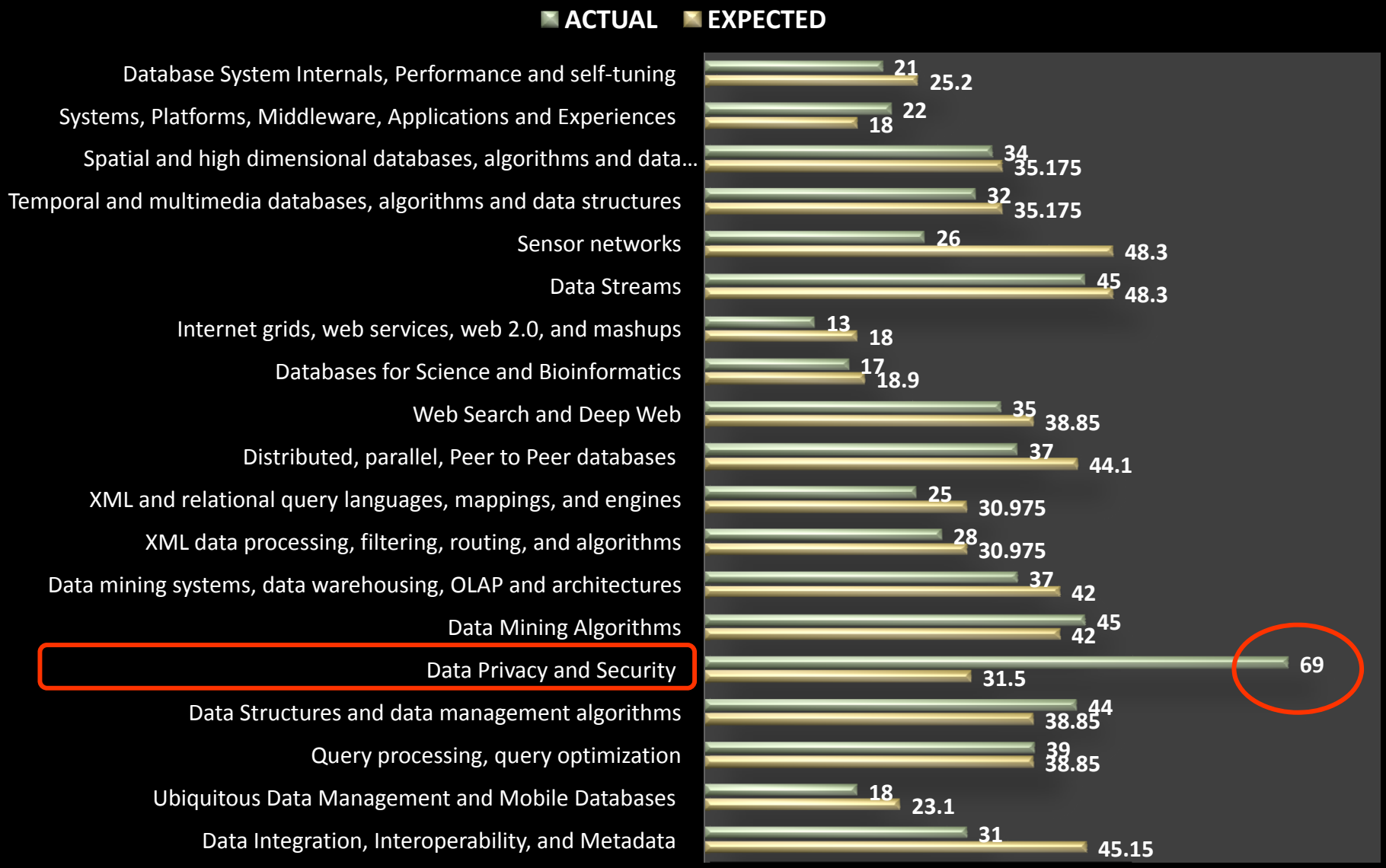


国際会議の様子
Research Track Submissions



	93	94	95	96	97	98	99	2000	2001	2002	2003	2004	2005	2006	2007	2008
ICDE	286	270	233	264	240	252	272	287	300	287	378	441	521	460	620	615
SIGMOD	235	279	208	290	202	246	205	248	293	240	342	431	431	446	480	435
VLDB	311	293	297	340	335	327	390	351	339	468	457	504	517	626	538	571

ICDE2008 での投稿分野



データ工学とセキュリティ

- プライバシィ・セキュリティの関心が高まっている
  - Security in Databases… 1970年代から
  - 国際会議の論文数が増え始めたのは 1990年代後半から
- 保護の対象
  - データベース（リレーション、属性、タプル、アイテム）
  - ストレージ（ブロック、ファイル）
- 保護する状況
  - 格納してあるデータ
  - 転送中のデータ
- DE分野でのいくつかのアプローチを紹介

あくまでもデータ工学側からの説明
データ工学からのセキュリティに
興味を持ち始めたのも最近
ぜひいろいろ教えてください！

Agenda

- データ工学とその動向
- 関係データベースとセキュリティ
- データマイニングとセキュリティ
- XMLデータベースとセキュリティ
- ストレージとセキュリティ
- ネットワークストレージとセキュリティ
  - 我々の取り組み
- まとめ

関係データベースの用語

名前	年齢	性別	身長	体重
太郎	30	男	175	65
次郎	24	男	180	75
三郎	20	男	170	60
花子	27	女	150	45
良子	20	女	160	50

スキーマ

属性

タプル

関係
（リレーション）

σ

Selection
タプルの選択

名前	年齢	性別	身長	体重
太郎	30	男	175	65
花子	27	女	150	45

π

Projection
属性の切出し

名前	年齢
太郎	30
次郎	25
三郎	20
花子	27
良子	20

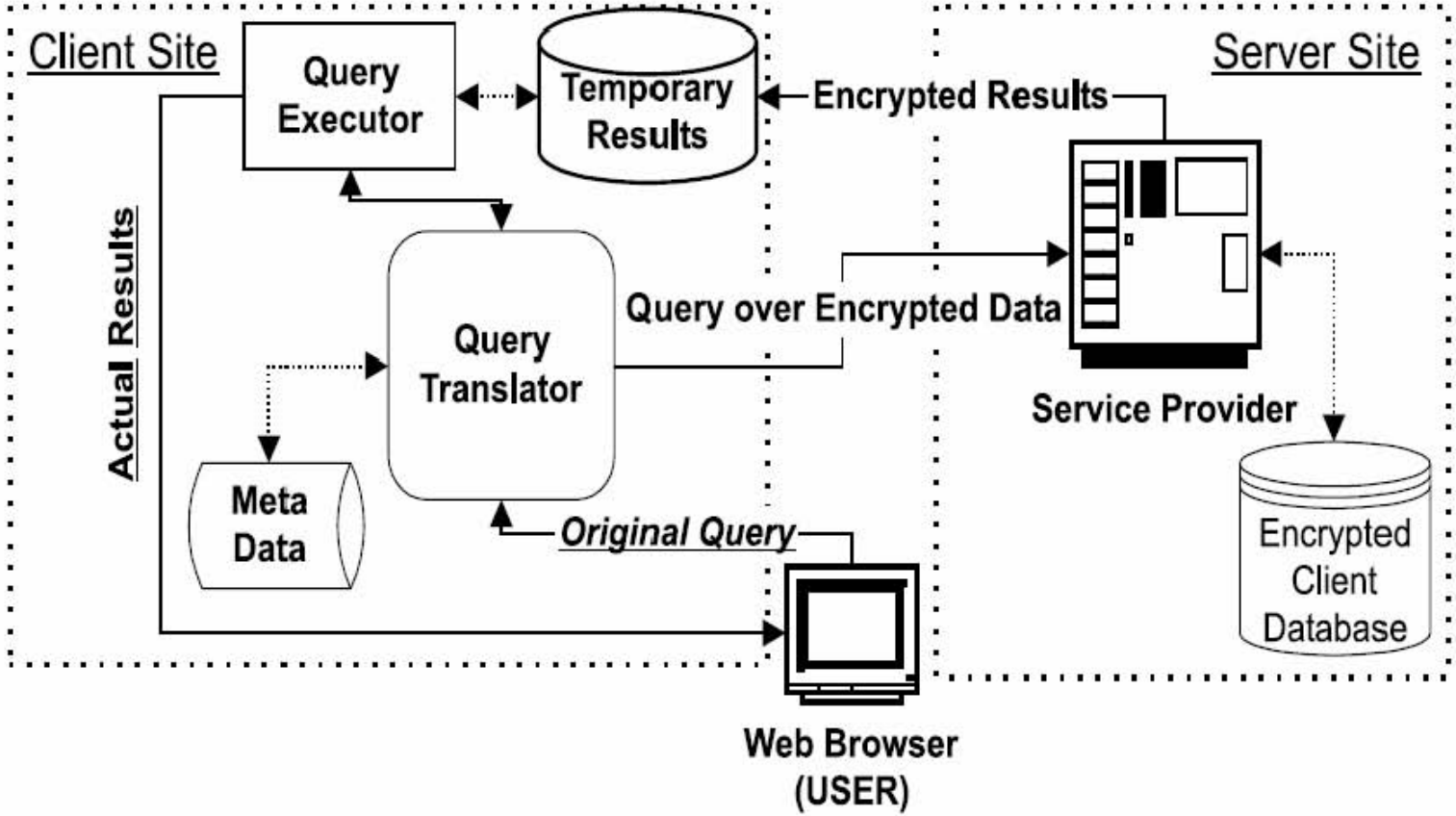
⋈

Join
関係の結合

名前	年齢	性別	身長	体重	区分	注意
太郎	30	男	175	65	A	○
花子	27	女	150	45	B	○

Executing SQL over Encrypted Data in the Database-Service-Provider Model

(Hacigumus et al. UC Irvine & IBM, SIGMOD 2002)



属性の暗号化

オリジナルのリレーション（テーブル）

eid	ename	salary	addr	did
23	Tom	70K	Maple	40
860	Mary	60K	Main	80
320	John	50K	River	50
875	Jerry	55K	Hopewell	110

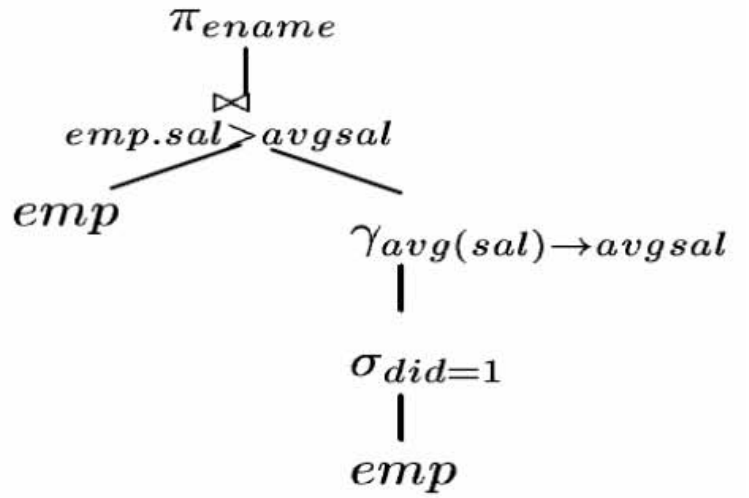
暗号化したリレーション（テーブル）：選択のための値域のマップ

etuple	eid^S	$ename^S$	$salary^S$	$addr^S$	did^S
1100110011110010...	2	19	81	18	2
1000000000011101...	4	31	59	41	4
1111101000010001...	7	7	7	22	2
1010101010111110...	4	71	49	22	4

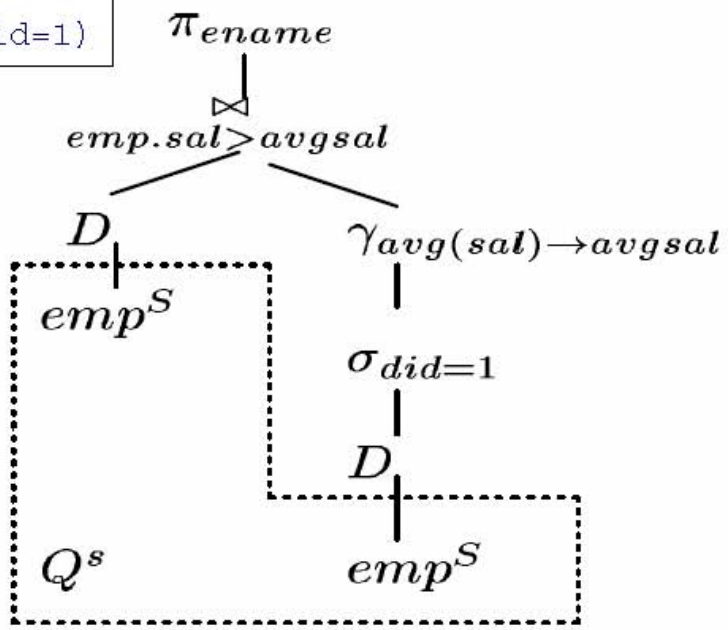
Hacigumus et al. (SIGMOD 2002)

オリジナルの問い合わせ木

```
SELECT emp.name FROM emp
WHERE emp.salary >
  (SELECT AVG(salary) FROM emp WHERE did=1)
```



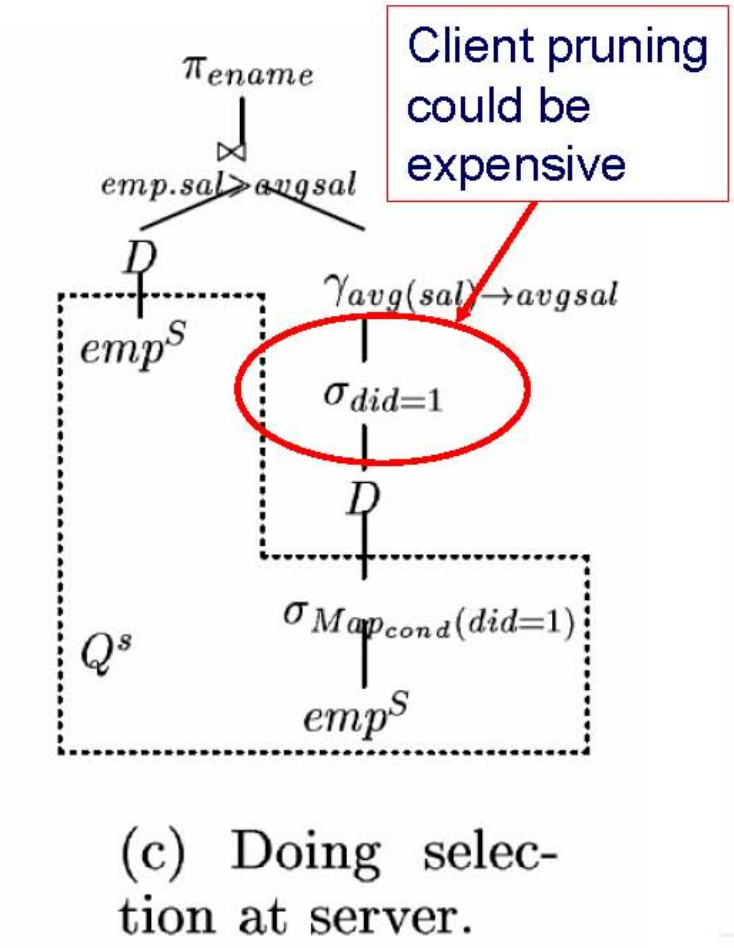
(a) Original query tree.



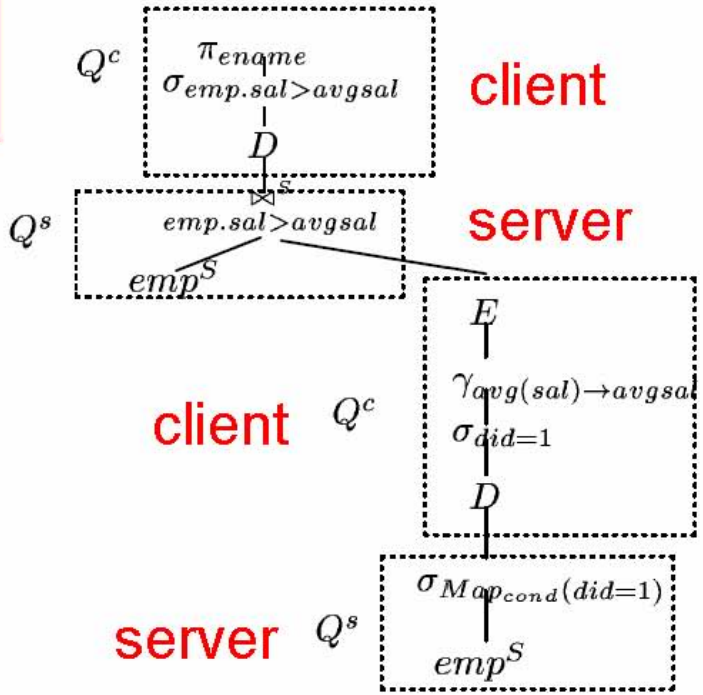
(b) Replacing encrypted relations.

Hacigumus et al. (SIGMOD 2002)

できるだけサーバー上で実行



(c) Doing selection at server.



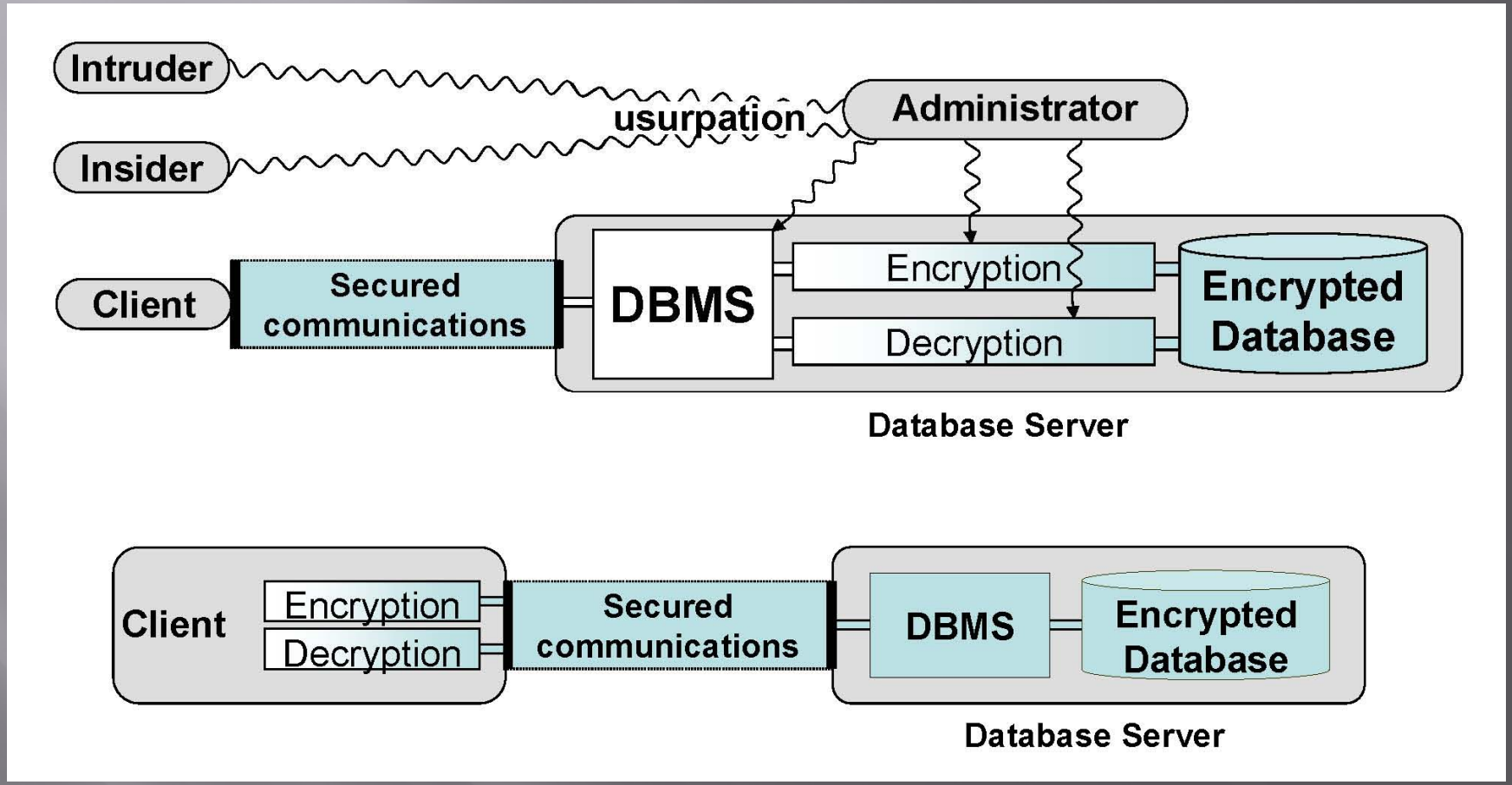
(d) Multiple interactions between Client and Server.

Hacigumus et al. (SIGMOD 2002)

Chip-Secured Data Access: Confidential Data on Untrusted Servers

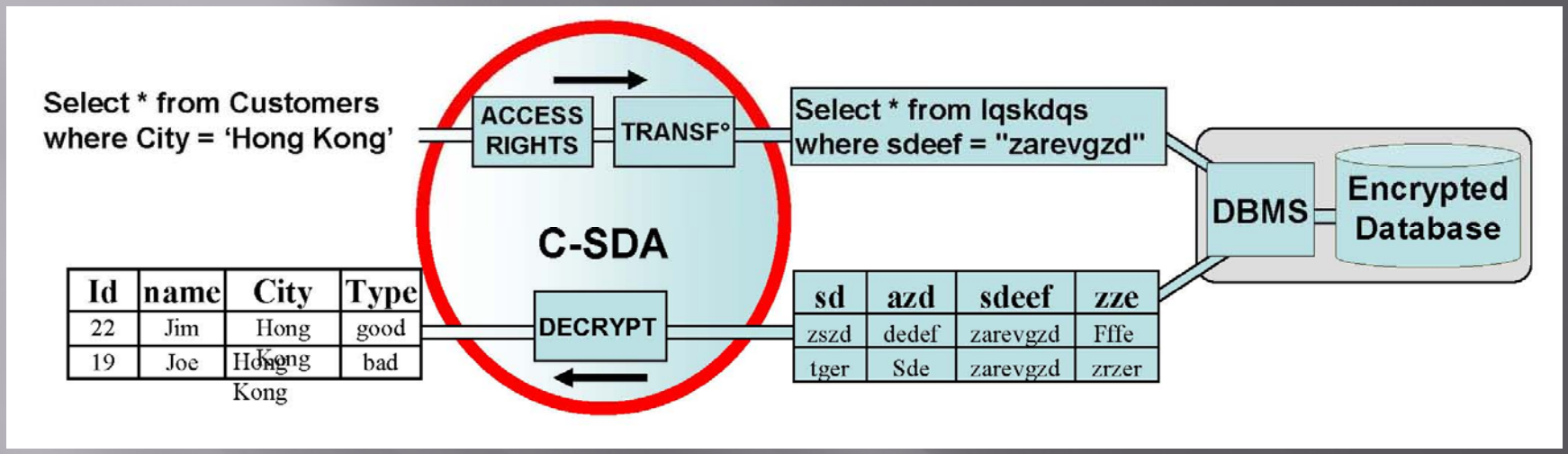
- Luc Bouganim and Philippe Pucheral
  - PRISM Laboratory , France
- VLDB2002
- C-SDA (Chip-Secured Data Access)
  - A mediator between a client and an encrypted DB
    - Security software embedded in a SmartCard
  - Secure communication betwenn them

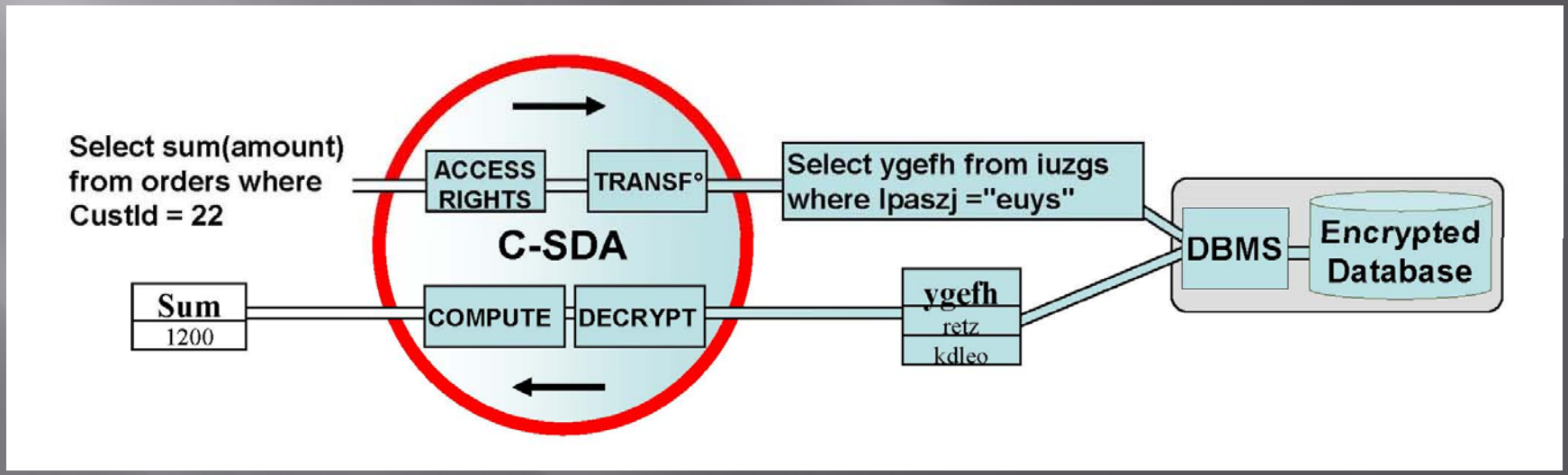
DB処理における機密保護の範囲



Bouganim et al. (VLDB 2002)

暗号化の対象





Bouganim et al. (VLDB 2002)

Agenda

- データ工学とその動向
- 関係データベースとセキュリティ
- データマイニングとセキュリティ
- XMLデータベースとセキュリティ
- ストレージとセキュリティ
- ネットワークストレージとセキュリティ
  - 我々の取り組み
- まとめ

Privacy Preserving Data Mining

- データベースをマイニングする際のプライバシィ
  - 元のデータが見えないだけでは不十分
- 非常に簡単な例：次ページのリレーション
  - 各学生の Grade Average は見えない
  - 統計結果はオープン
  - 攻撃者はCarol がCSの女子学生であることを知っている
  - CS の女子学生が一人なので
    - Q1: SELECT Count (*) FROM Students WHERE Sex =‘F’ AND Programme = ‘CS’
    - Q2: SELECT Avg (Grade Ave) FROM Students WHERE Sex =‘F’ AND Programme = ‘CS’

    で Carol の Grade Average がわかる

Inference - Example

Name	Sex	Programme	Units	Grade Ave
Alma	F	MBA	8	63
Bill	M	CS	15	58
Carol	F	CS	16	70
Don	M	MIS	22	75
Errol	M	CS	8	66
Flora	F	MIS	16	81
Gala	F	MBA	23	68
Homer	M	CS	7	50
Igor	M	MIS	21	70

Agenda

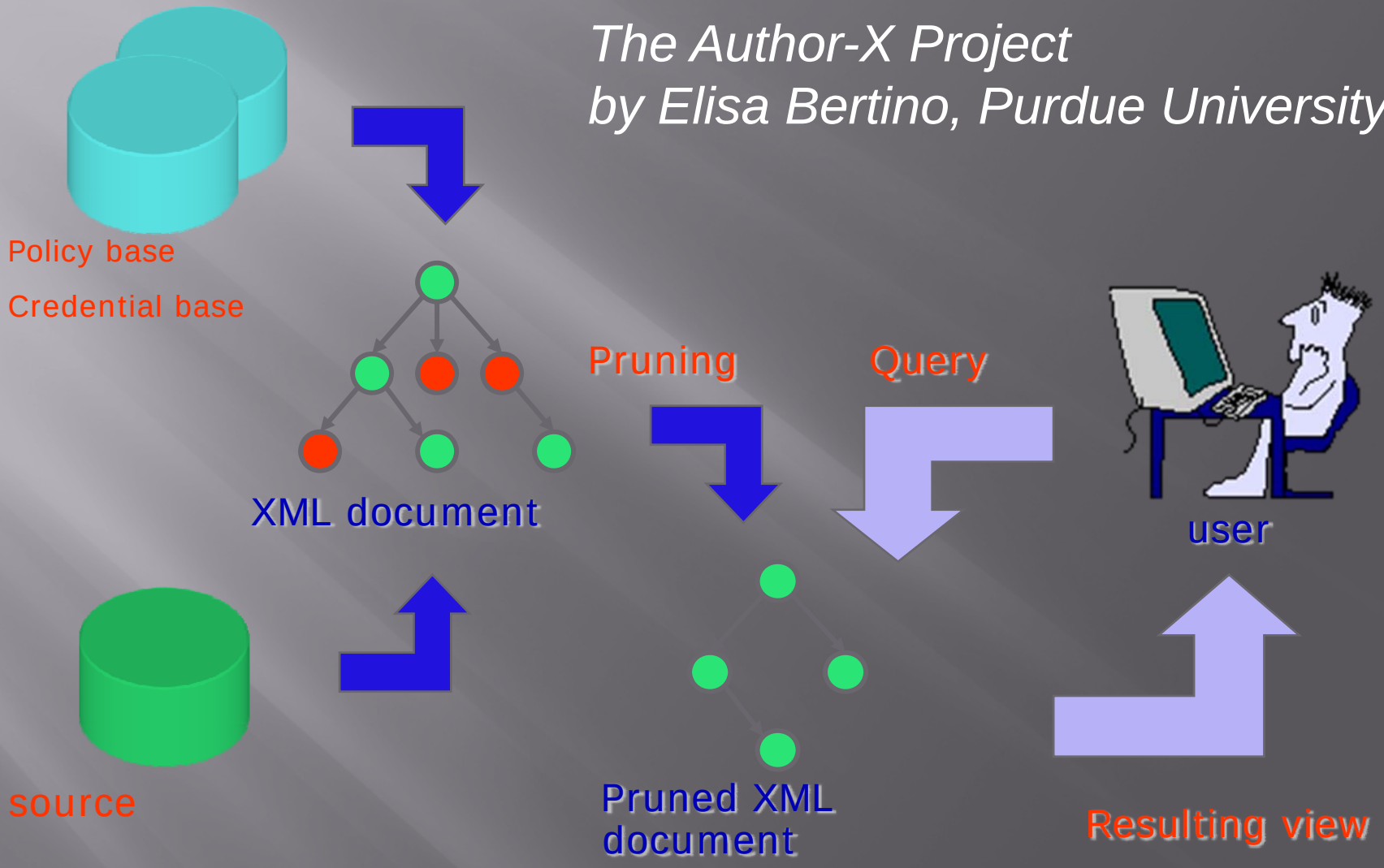
- データ工学とその動向
- 関係データベースとセキュリティ
- データマイニングとセキュリティ
- XMLデータベースとセキュリティ
- ストレージとセキュリティ
- ネットワークストレージとセキュリティ
  - 我々の取り組み
- まとめ

XML データベースとセキュリティ

- XML: Extensible Markup Language
  - タグで階層構造（論理構造）を埋め込む
  - 全体として木構造を構成
- XMLデータベース
  - 属性名とタグ名を対応
  - 関係データベースより柔軟に情報を表現
    - 属性間の関係を表現でき、場合によって変更可能
- XML データベースのセキュリティ
  - 単なる属性単位ではなく、その階層構造を考慮する必要
  - ポリシィベースでアクセス制御
    - ポリシィ言語：XACL (XML Access Control Language)

XML Access Control

The Author-X Project
by Elisa Bertino, Purdue University



1つの XML 木を異なるキーで暗号化：アクセス可能な部分だけ抽出

```
<WorldLawBulletin Date="8/8/1999">
 <Law Country="USA" RelatedLaws = "LK75"/>
     <Topic>Taxation</Topic> <Summary>...</Summary>
  </Law>
 <Law Id="LK75" Country="Italy"/>
     <Topic>Import-Export</Topic> <Summary>...</Summary>
  </Law>
 <BluePageReport>
    <Section GeoArea="Europe">
      <Law Country="Germany"/>
        <Topic>Guns</Topic> <Summary>...</Summary>
      </Law>
      ...
    </Section>
    <Section GeoArea="NorthAmerica">
      <Law Country="USA"/>
        <Topic>Transportation</Topic> <Summary>...</Summary>
      </Law>
      ...
    </Section>
 </BluePageReport>
</WorldLawBulletin>
```

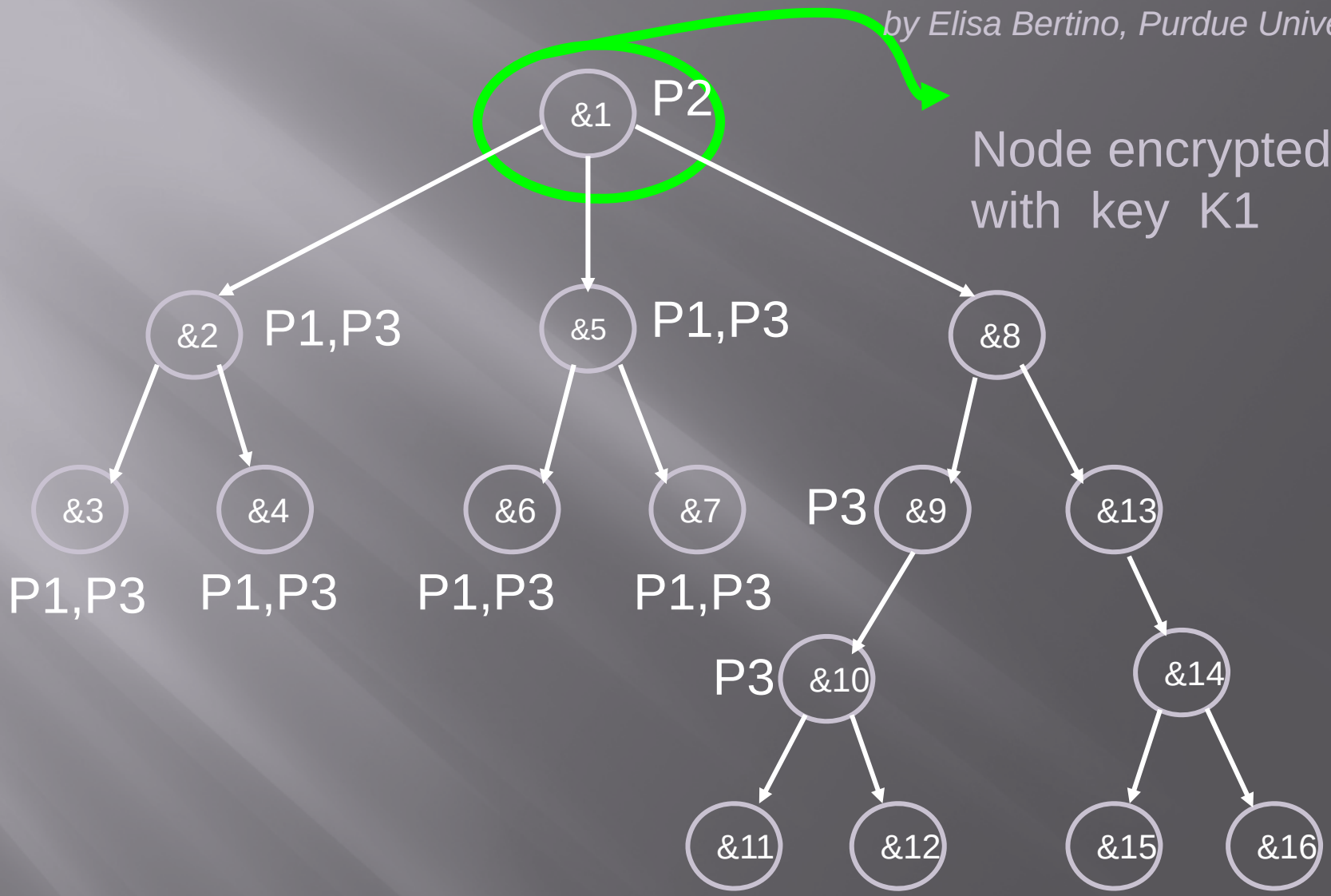
An XML Document

XMLの木構造

by Elisa Bertino, Purdue University

Well-Formed Encryption

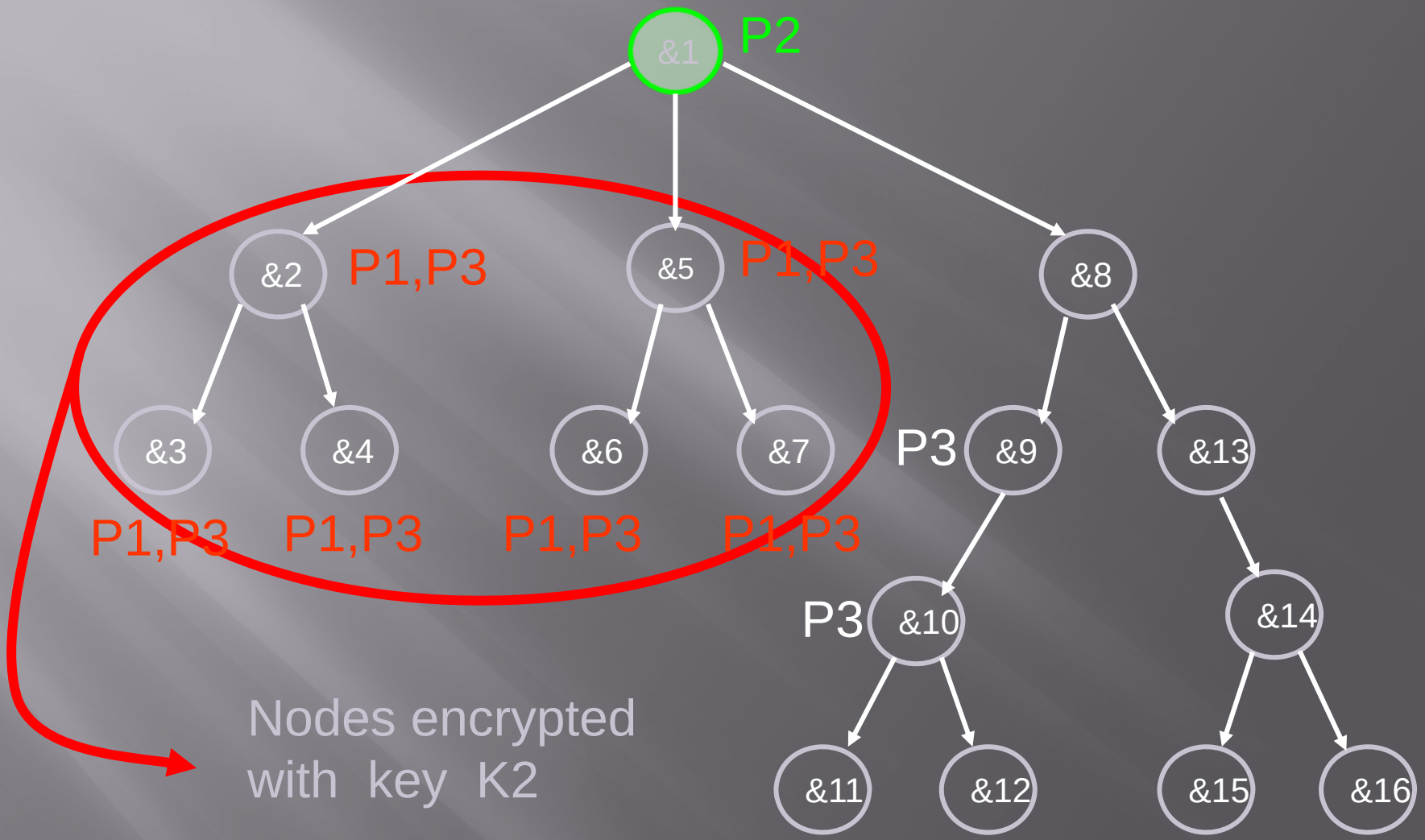
by Elisa Bertino, Purdue University



ポリシーに合わせてキーを割り当て

Well-Formed Encryption

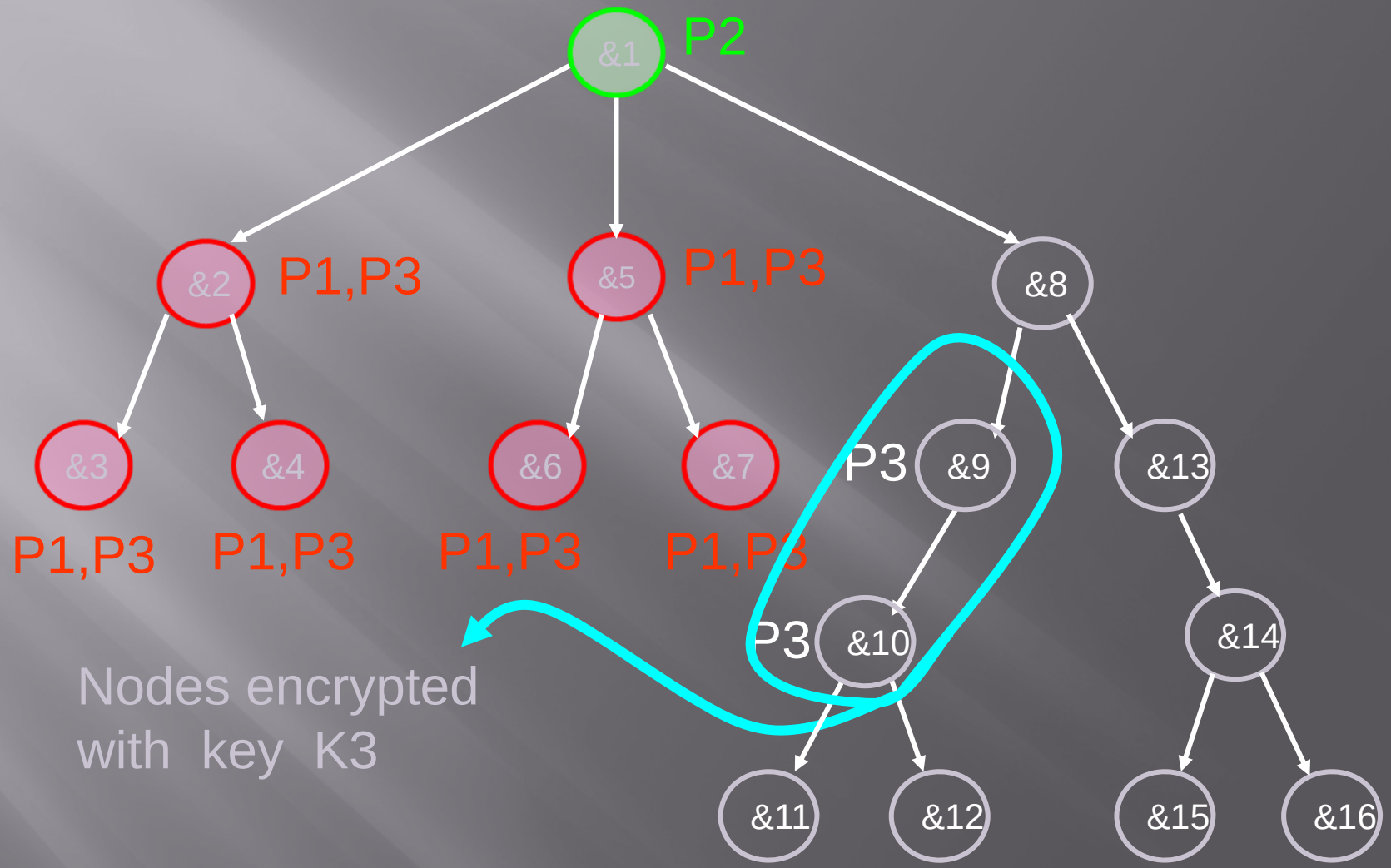
by Elisa Bertino, Purdue University



ポリシーの重なりにも対応

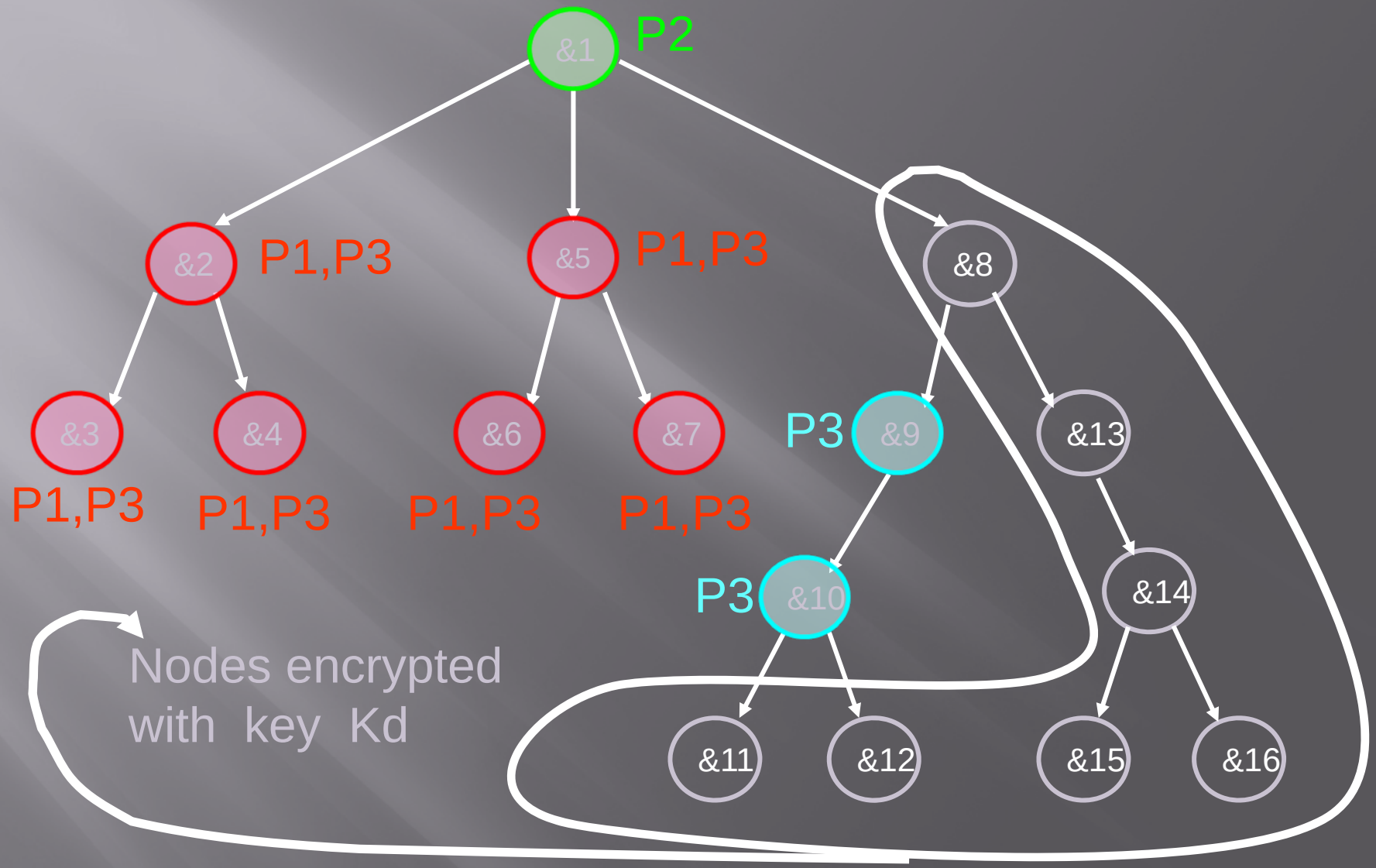
Well-Formed Encryption

by Elisa Bertino, Purdue University



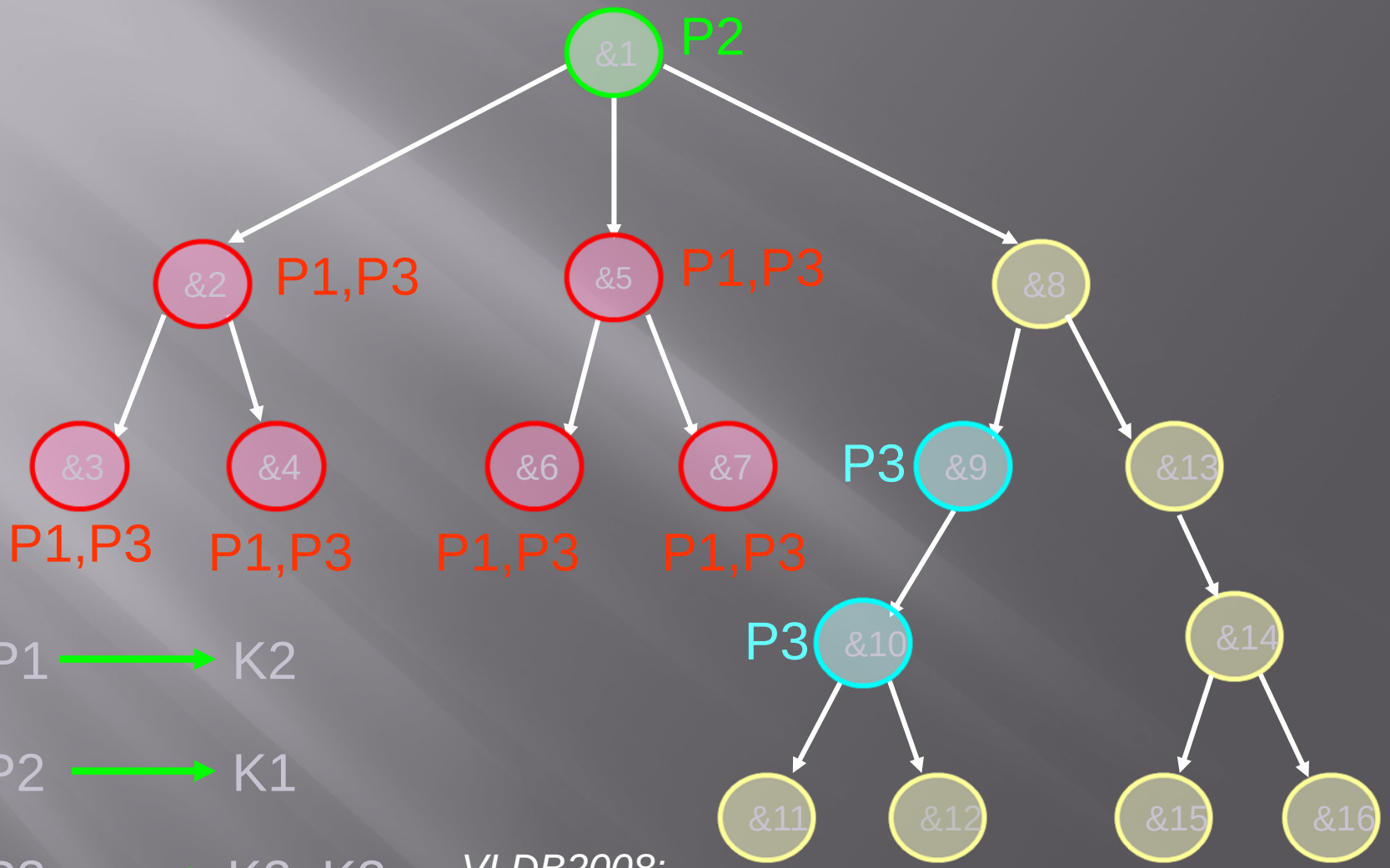
Well-Formed Encryption

by Elisa Bertino, Purdue University



Well-Formed Encryption

by Elisa Bertino, Purdue University



P1 → K2

P2 → K1

P3 → K2, K3

VLDB2008:
Structural Signatures for Tree Data Structures
Ashish Kundu, Elisa Bertino (Purdue University, USA).

Agenda

- データ工学とその動向
- 関係データベースとセキュリティ
- データマイニングとセキュリティ
- XMLデータベースとセキュリティ
- ストレージとセキュリティ
- ネットワークストレージとセキュリティ
  - 我々の取り組み
- まとめ

ストレージのセキュリティ

- より一般にストレージのレベルでセキュリティ制御
  - DB だけでなくファイルも
- アプローチ
  - ATA セキュリティコマンド
    - ATA I/F を介してアクセスを制限するのみ、平文で記録
  - ソフト暗号化
    - OS の基本ドライバ/アプリケーションとして暗号化
  - ハード暗号化
    - HDD 内でハードウェア（LSI）で暗号化
- ストレージのセキュリティの動向
  - ハード暗号化：シーゲート、富士通

セキュアストレージアプローチ比較

方式	アクセス制限	耐メディア解析	耐メモリ解析	性能低下
ATAコマンド	○	×	-	なし
ソフト暗号化	○	○	×	あり
ハード暗号化	○	○	○	なし

- ATAセキュリティコマンド
  - SECURITY SET PASSWORD
  - SECURITY UNLOCK
  - SECURITY ERASE PREPARE
  - SECURITY ERASE UNIT
  - SECURITY FREEZE LOCK
  - SECURITY DISABLE PASSWORD
- メディア解析
  - 記録メディアを取り出して内容を読みだす
- メモリ解析（Cold Boot Atack）
  - メモリ（DRAM）中に残った暗号キーを取得

Cold Boot Attacks on Encryption Keys

USENIX Security Sympo. Princeton Univ. 2008.2

J. Alex Halderman et al.

Cold Boot Attacks on Encryption Keys

USENIX Security Sympo. Princeton Univ. 2008.2

J. Alex Halderman et al.

Cold Boot Attacks on Encryption Keys

USENIX Security Sympo. Princeton Univ. 2008.2

J. Alex Halderman et al.

Agenda

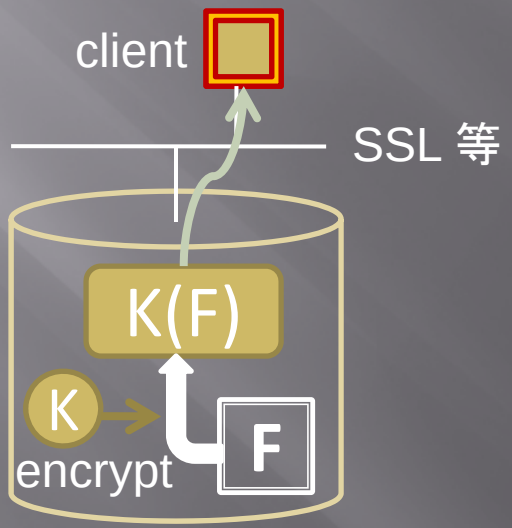
- データ工学とその動向
- 関係データベースとセキュリティ
- データマイニングとセキュリティ
- XMLデータベースとセキュリティ
- ストレージとセキュリティ
- ネットワークストレージとセキュリティ
  - 我々の取り組み
- まとめ

ネットワークストレージのセキュリティ

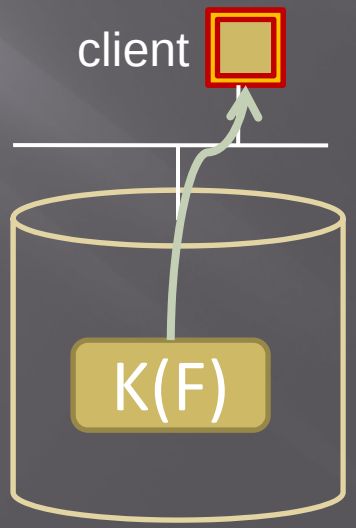
- ストレージシステムの動向
  - サーバー毎のストレージ　→　ネットワークストレージ
  - ストレージコンソリデーションによる管理コストの削減
- ネットワークストレージにおける機密性
  - **通信路における機密性**
    - アクセス権を持たないユーザに転送中のデータを傍受されても読めない
  - **ノードにおける機密性**
    - 攻撃者による不正ログイン
    - ノードの陥落時にデータが漏洩しない（含むメディア解析）

ネットワークストレージの機密保持

- 伝送中データの機密性保持の為の暗号利用方式
  - Encrypt-On-Wire方式
    - 平文で格納し，転送時に暗号化（e.g. SSL）
  - Encrypt-On-Disk方式
    - データを暗号化して格納し，転送時はそのまま送信
- Encrypt-On-Disk方式の方がデータ転送性能が高い
  - ストレージ側送受信時に暗号処理が不要のため
- Encrypt-On-Disk方式ではノード上の機密性保持も実現可能



Encrypt-On-Wire



Encrypt-On-Disk

Encrypt-On-Diskの問題点

- アクセス権失効（Revocation）に伴い再暗号化が必要
  - 権限を失ったユーザが暗号鍵を保持したままの場合
    - 伝送データの傍受等でデータが漏洩してしまう
- 既存の再暗号化手法
  - Active Revocation : Revocation発生時直ちに再暗号化
    - セキュリティ面で優れる
    - 性能面に問題
      - いかなる場合も直ちに再暗号化するためアクセスをブロック
      - 集中して発生するとリソースを圧迫
  - Lazy Revocation : 次の更新まで再暗号化処理を遅延
    - パフォーマンス面で優れる
      - 複数のRevocationに対する処理を一度の更新で一括処理
    - 古い鍵で暗号化された脆弱なファイルが残る

➔ 性能面とセキュリティ面のトレードオフ

我々の取り組み

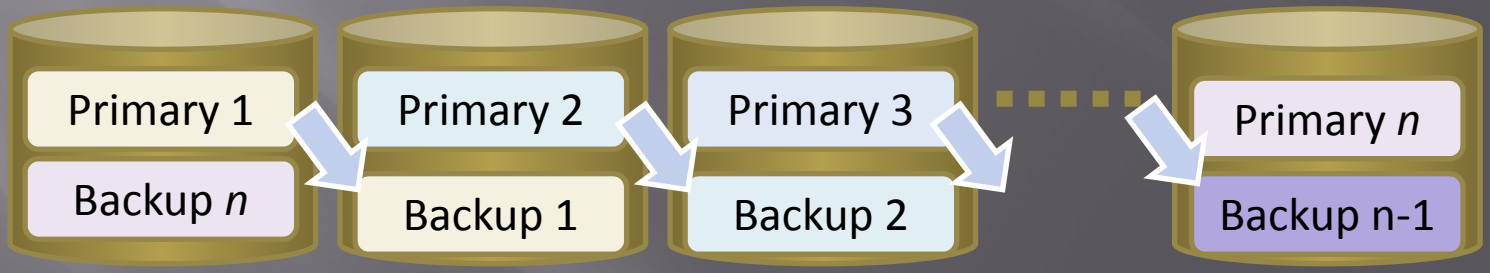
- ディペンダブルなストレージシステムの実現
  - リライアブル（高信頼）かつセキュア
- ディペンダビリティ
  - 従来の耐故障化だけでなくセキュリティの面も併せ持つ
    - 耐故障：**偶然の故障**に耐える
    - セキュリティ：**故意の攻撃**に耐える
- これまでは耐故障に焦点を当ててきた
  - これからはセキュリティも対象に
- ストレージの耐故障化
  - RAID ：スペース効率が良いが、故障時、回復時の性能劣化
  - プライマリ・バックアップ：スペース効率は悪いが性能劣化少

我々の方針

- 目的
  - Encrypt-On-Disk方式の高信頼な分散ストレージにおける性能とセキュリティの両立・向上
- アプローチ
  - Active Revocationを採るEncrypt-On-Disk方式を採用
  - 処理のコストを削減することで，性能とセキュリティを両立した高信頼ストレージシステムを目指す
    - Revocation発生時の性能，セキュリティを両立させる
      - Active Revocationの性能面を向上させることにより，セキュリティを維持しつつ，Revocation発生時の性能向上を図る
    - 再暗号化処理等をストレージ側で処理しユーザの負担を軽減しつつ，ディスクノード上のセキュリティを実現

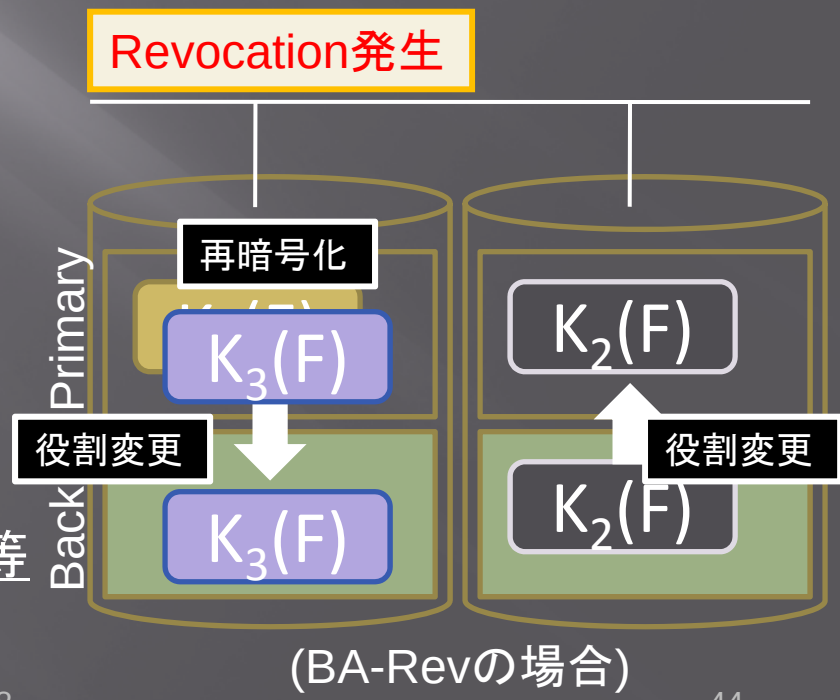
システムの前提

- 高機能ストレージ（e.g. 自律ディスク [Yokota, 1999]）
  - ストレージをインテリジェント化
    - 負荷分散や故障回復等，様々な処理をストレージ側で自律的に行うことで管理者の負担を軽減
  - ➔ 再暗号化処理をストレージ側で行う
    （鍵は各ユーザの公開鍵で暗号化されディスク上で管理し，該当ユーザのみ獲得可）
- プライマリ・バックアップ構造（高信頼化）
  - アクセスされるPrimaryデータと，その複製のBackupデータからなり，耐故障化のために異なるストレージノードに配置
  - 例：Chained Declustering
    - 隣接ノードにBackupを配置



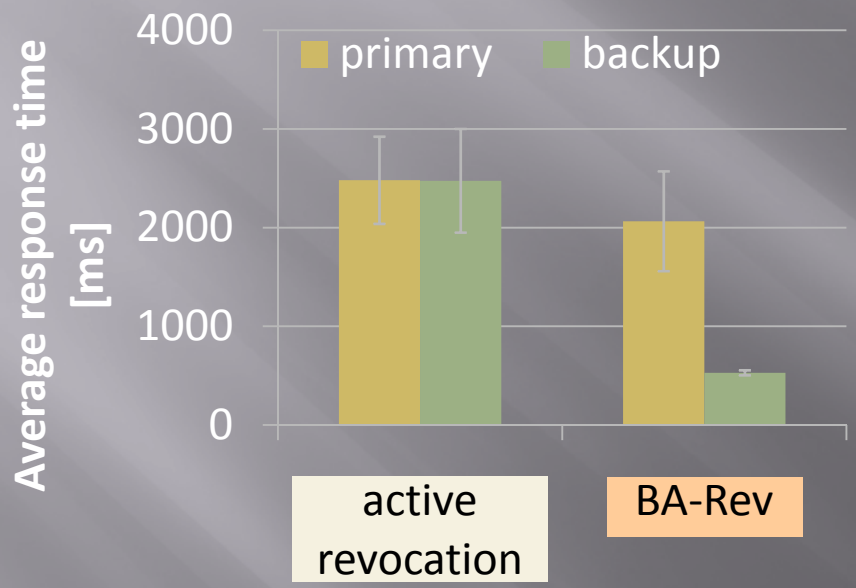
BA-Rev: Backup Assist Revocation

- 課題
  - Active Revocationはセキュリティ面で優れるが，アクセス遅延やリソース不足を引き起こす可能性
    - ➔セキュリティを保ちつつ，高性能なRevocationを実現したい
- 提案
  1. アクセスされないBackupを予め新しい鍵で暗号化しておき，Revocationに伴い役割を変更
  2. 元のPrimaryを再暗号化してBackupに設定
  - 実行タイミングはActive Revocationと同じ
    - ➔ セキュリティはActive Revocationと同等

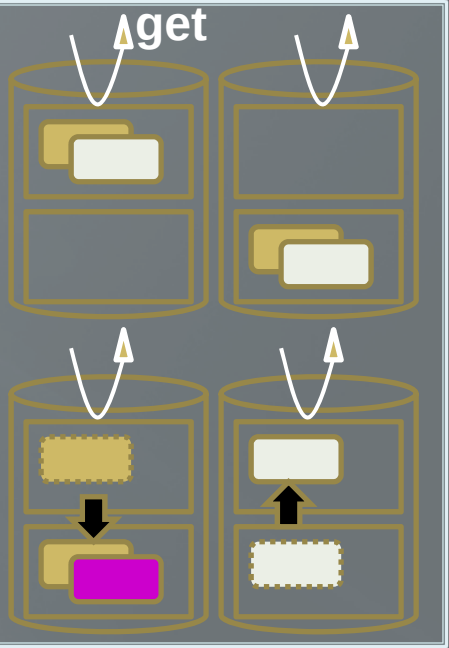


(BA-Revの場合)

BA-Rev の効果



- Active Revocation
  - Primary, Backup 共に再暗号化
- BA-Rev
  - Primary: 役割変更後再暗号化
  - Backup: 役割変更のみ

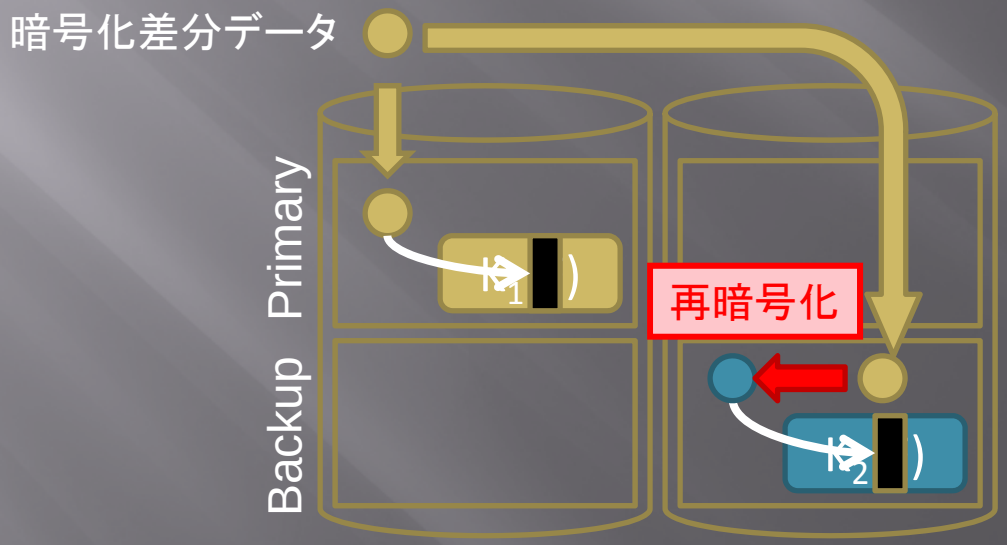


- ファイルの読み出し操作（get）のみの環境では，Active Revocationと比較して，Revocation発生前後の平均応答時間が改善
  - 新しいPrimaryへ再暗号化処理時間を待たずにアクセス可
  - 旧Backup側では（非常に処理が重い）再暗号化処理が不必要な為，性能低下を招かない

性能面での課題

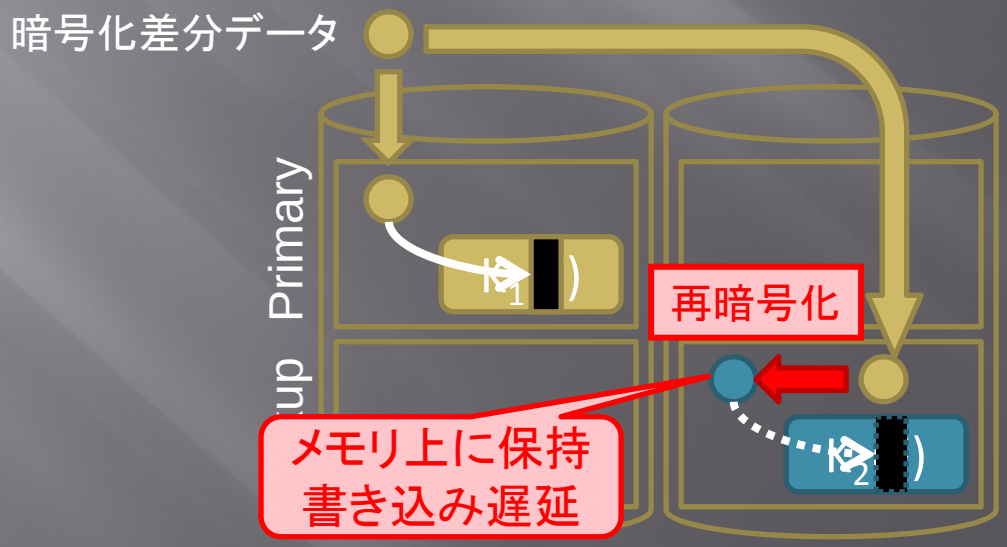
- 更新（update）時に性能低下の可能性
  - PrimaryとBackupで鍵が異なる為，Backup側で差分データの適用時に再暗号化が必要

➔再暗号化処理が重く，性能低下の恐れ



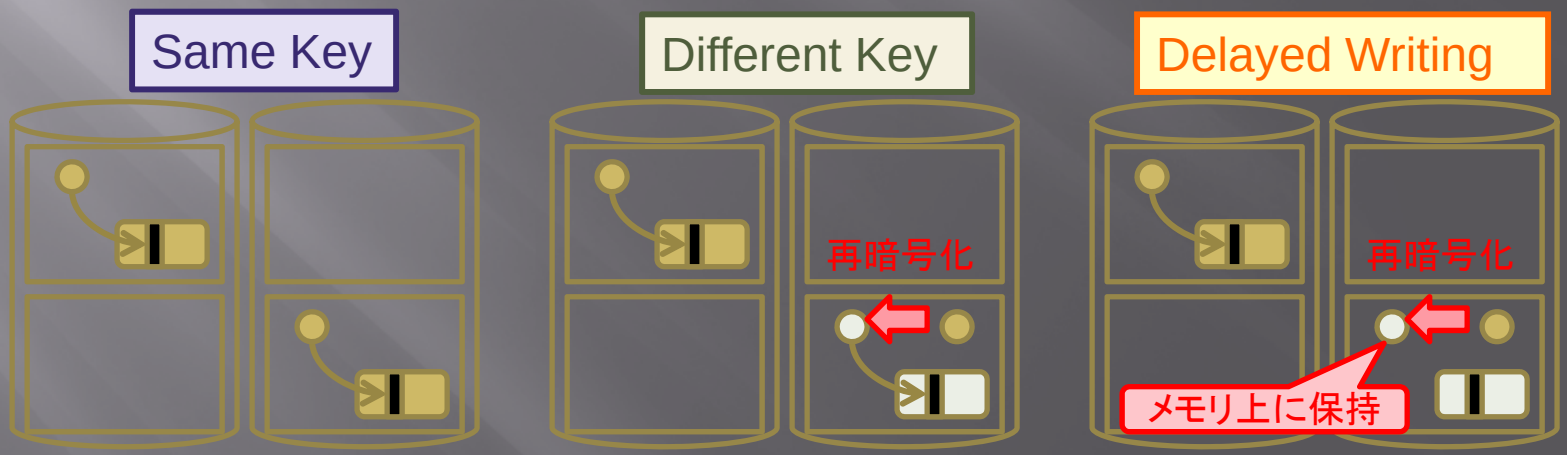
Delayed Writing

- Backup側において，差分データの再暗号化処理と書き込み処理を分割し，ファイルへの適用を遅延
  - BA-Revの異なる鍵で暗号化されたファイルの更新による性能劣化を改善



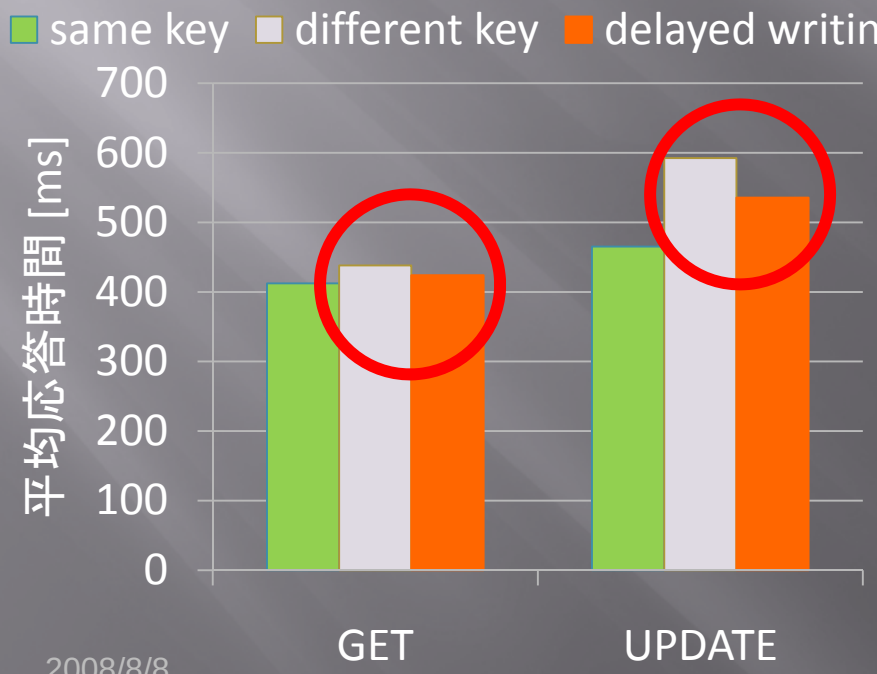
Delayed Writingの性能評価(1/3)

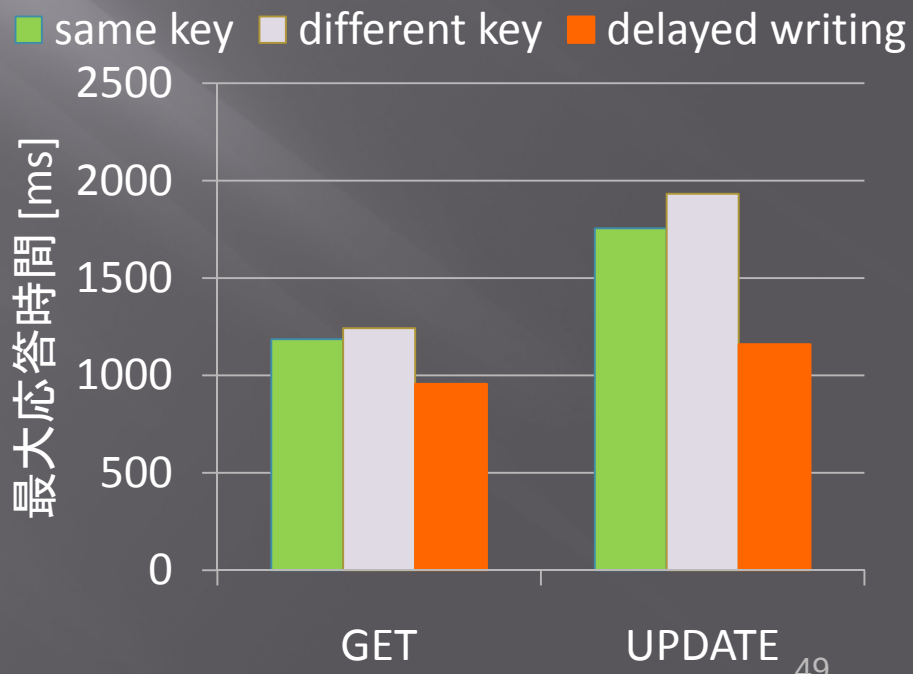
- 以下の3環境において，get･updateの応答時間を測定
  - Same Key: PrimaryとBackupは同じ鍵で暗号化
  - Different Key: PrimaryとBackupは異なる鍵で暗号化
  - Delayed Writing: Different Keyに提案手法を適用
- update時の処理
  - getとは異なりBackupでも処理が発生



Delayed Writing の性能評価(2/3)

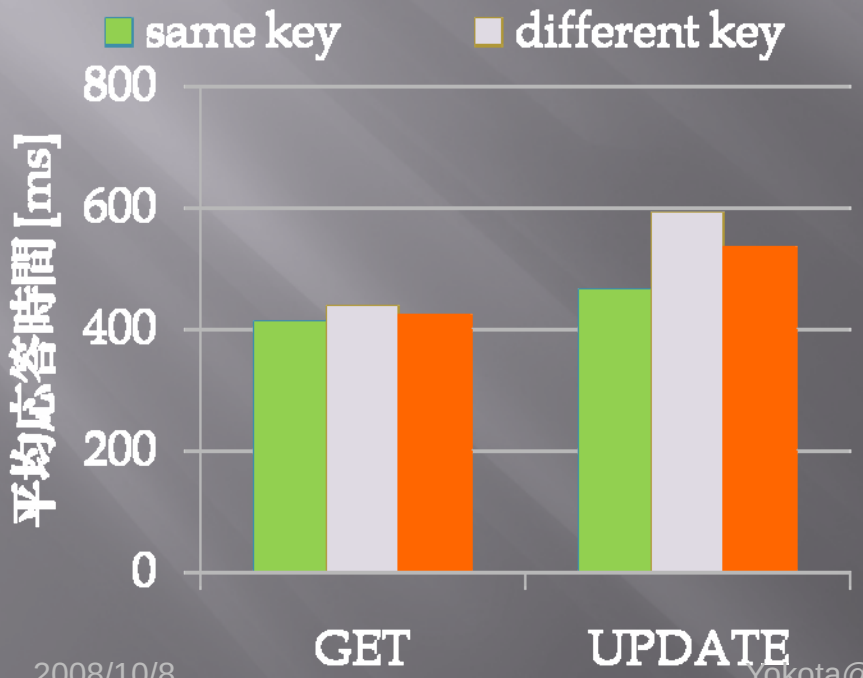
- Different Keyと比較して，Delayed Writingでは平均応答時間が改善
  - 差分適用遅延による効果

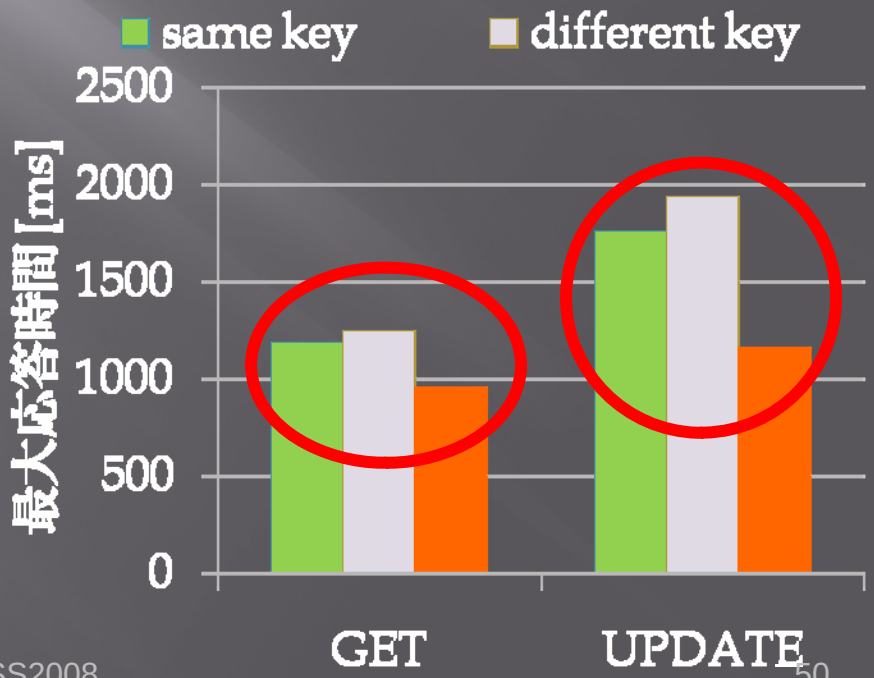




Delayed Writing の性能評価(3/3)

- 最大応答時間はDelayed Writingが最も良い結果に
  - Delayed WritingではBackupのプロセスでディスクI/Oが無い
  - 他方式ではディスクI/Oがあり，Primaryプロセスの I/Oと重なるとブロックされ，応答が大きく遅れる場合がある





Delayed Writing の信頼性

- Delayed Writingでは差分データの適用を遅延し，メモリ上に保持

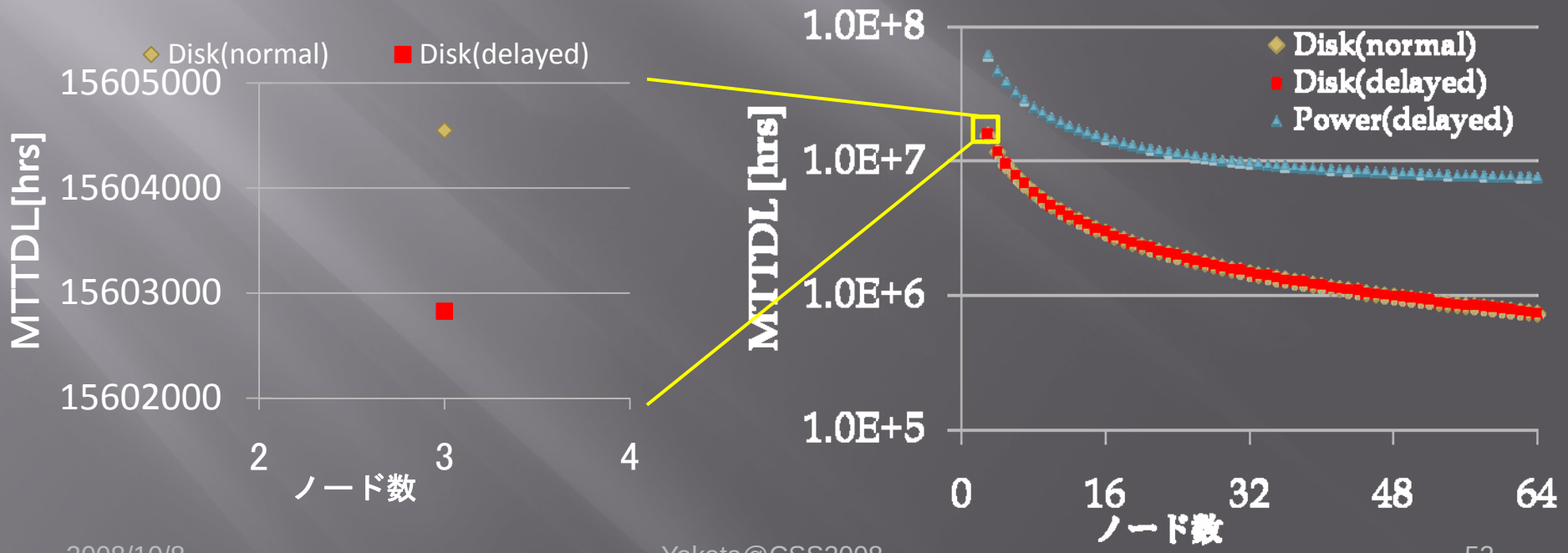
➔ 通常の環境よりデータ喪失の可能性が増加

1. 差分未適用のため，ノード故障時のMTTR（平均復旧時間）が増加
   - ➔冗長性回復前にディスクが故障してデータが失われる可能性も増大
2. 電源系とディスクが同時に故障することでデータ喪失

- 信頼性低下の程度を評価する為，ディスク故障，電源系故障(Delayed Writingのみ)に関係するMTTDL（平均データ喪失時間）を算出し比較

信頼性評価

1. ディスク故障に関するMTTDLの差は非常に小さい
   - MTTR増大による信頼性低下は小さい
2. 提案手法の電源系故障によるMTTDLは非常に大きく，全体への影響は小さい



セキュリティ面での課題

- 通常の再暗号化は暗号化と復号が不可逆
  - 途中で平文を生成
  1. 元の暗号鍵で復号
  2. 新しい暗号鍵を生成し暗号化

→ ストレージ上で再暗号化処理を実行すると，ノード上の機密性を実現できない

- ネットワークストレージのノードは攻撃者により陥落し，内部データを盗み見られる可能性がある
- 再暗号化処理中に平文が出現するため，平文が漏洩する危険性がある

途中で平文が生成されない再暗号化処理手法が必要

RORE: Reverse Order Re-Encryption

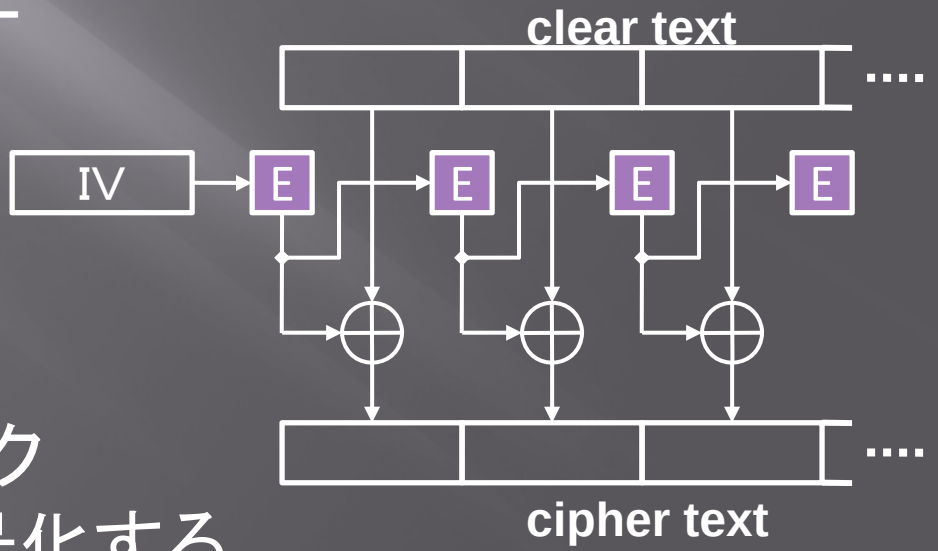
- 処理中に平文を生成しない再暗号化手法
  - 通常の再暗号化：一度平文に戻すのが必要

  - RORE：暗号化を先に行えるので平文が生成されない

- 暗号モードとしてOFBを用いることで再暗号化における復号と暗号化を可逆にできる

OFB(Output FeedBack)モード

- 初期ベクトル（Initial Vector: IV）を繰り返し暗号化して疑似乱数ビット列を生成し，平文との排他的論理和により暗号文を生成
- 復号処理は暗号化処理と同じで，暗号化に用いた疑似乱数ビット列と排他的論理和をとる
- ストレージに格納されたファイルをAES等のブロック暗号で，OFBモードで暗号化する



RORE の実現アプローチ

- 排他的論理和で交換則が成り立つことと，OFB の構造を利用し，再暗号化処理を暗号化，復号の順に実行
  - 暗号文＝平文⊕暗号ビット列1
  - 通常の再暗号化：
    （暗号文⊕暗号ビット列1）⊕ 暗号ビット列2
    （平文）
  - RORE：上記処理を以下のように変換
    （暗号文⊕暗号ビット列2）⊕ 暗号ビット列1
    （二重の暗号文）
- 処理中に平文が出現しない

➔ 再暗号化処理をストレージノード上で実行しても，そのノード陥落によるデータ漏洩の危険性が少ない

（K1(F)、K2(F)、K1K2(F) 全てが読みだせないという前提）

RORE での留意点

- 堅牢な鍵管理手法の必要性
  - ファイルと同じストレージ上で無防備に管理していると，RORE を用いても漏洩の危険がある
  - 鍵管理の考慮が必要
    - ファイルと異なるストレージで鍵を暗号化して管理（本実験での環境）
    - 堅牢な鍵サーバの設置
    - セキュリティソフト組込スマートカード
    - セキュアチップ
    - etc.

実験環境

- RORE評価の為，PCクラスタ上にEncrypt-On-Disk方式ファイルサーバクライアントプログラムを作成
- 以下の2環境を想定
  - Normal:
    - 復号，暗号化の順に再暗号化する通常の環境
    - ファイルと鍵を同ノードで管理
    - ノード上の機密性保障不可
  - RORE:
    - 提案手法の再暗号化を行う環境
    - 鍵をファイルと異なるノードで管理
    - ノード上の機密性を保証

CPU	AMD Athlon XP-M1800+(1.53GHz)
Memory	PC2100 DDR SDRAM 1GB
HDD	TOSHIBA MK3019GAX (30GB，5400rpm，2.5inch)
Network	TCP/IP + 1000BASE-T
OS	Linux 2.4.20
Java VM	Sun J2SE 1.5.0_03 Server VM

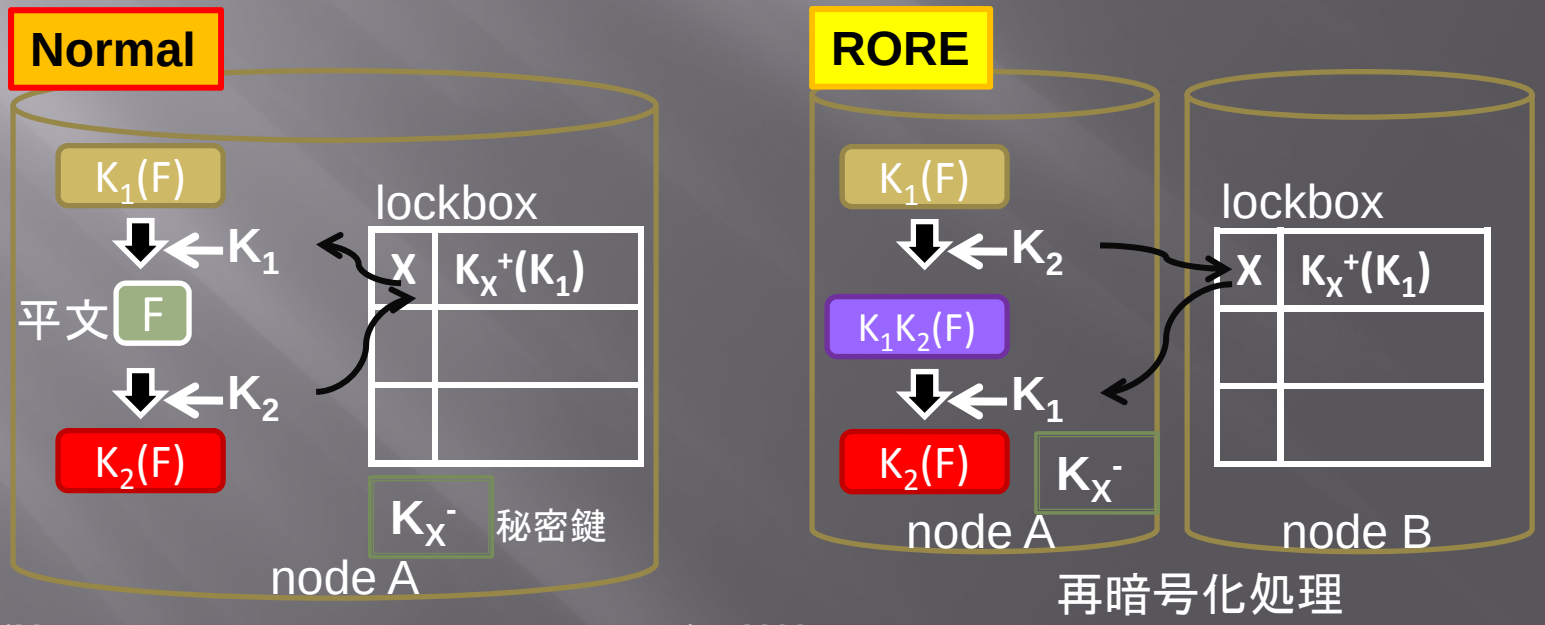
共通鍵暗号	AES 128bit
公開鍵暗号	RSA 1024bit
暗号化モード	OFB
パディング	なし
Zipf 母数θ	0.7
ストレージノード数	3
ファイルサイズ	1MB
ファイル数	500個/node

再暗号化の比較

- 鍵管理構造としてロックボックスを利用
  - ファイルの共通鍵(K_n)の使用権を持つユーザの公開鍵(K_x^+)で暗号化され格納
  - 使用権を持つユーザ（＝対応する秘密鍵をもつユーザ）のみ共通鍵を獲得可
- RORE ではファイルとは異なるノードにロックボックスを格納
  - 処理中のいかなる時点でも，1ノード内のデータから平文を生成不可

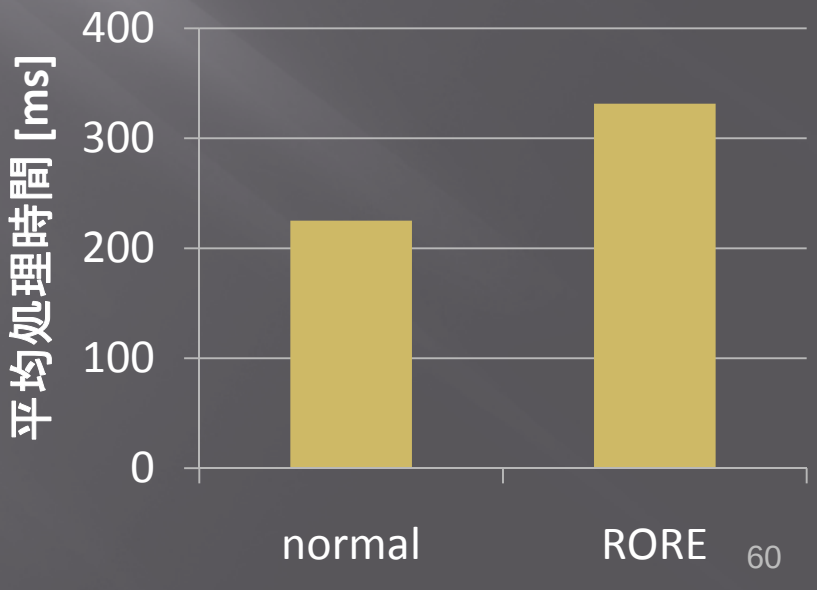
➔ 1ノード陥落に対して機密性を保証

  - ファイルの共通鍵の送受信は送信先の公開鍵で暗号化して行う



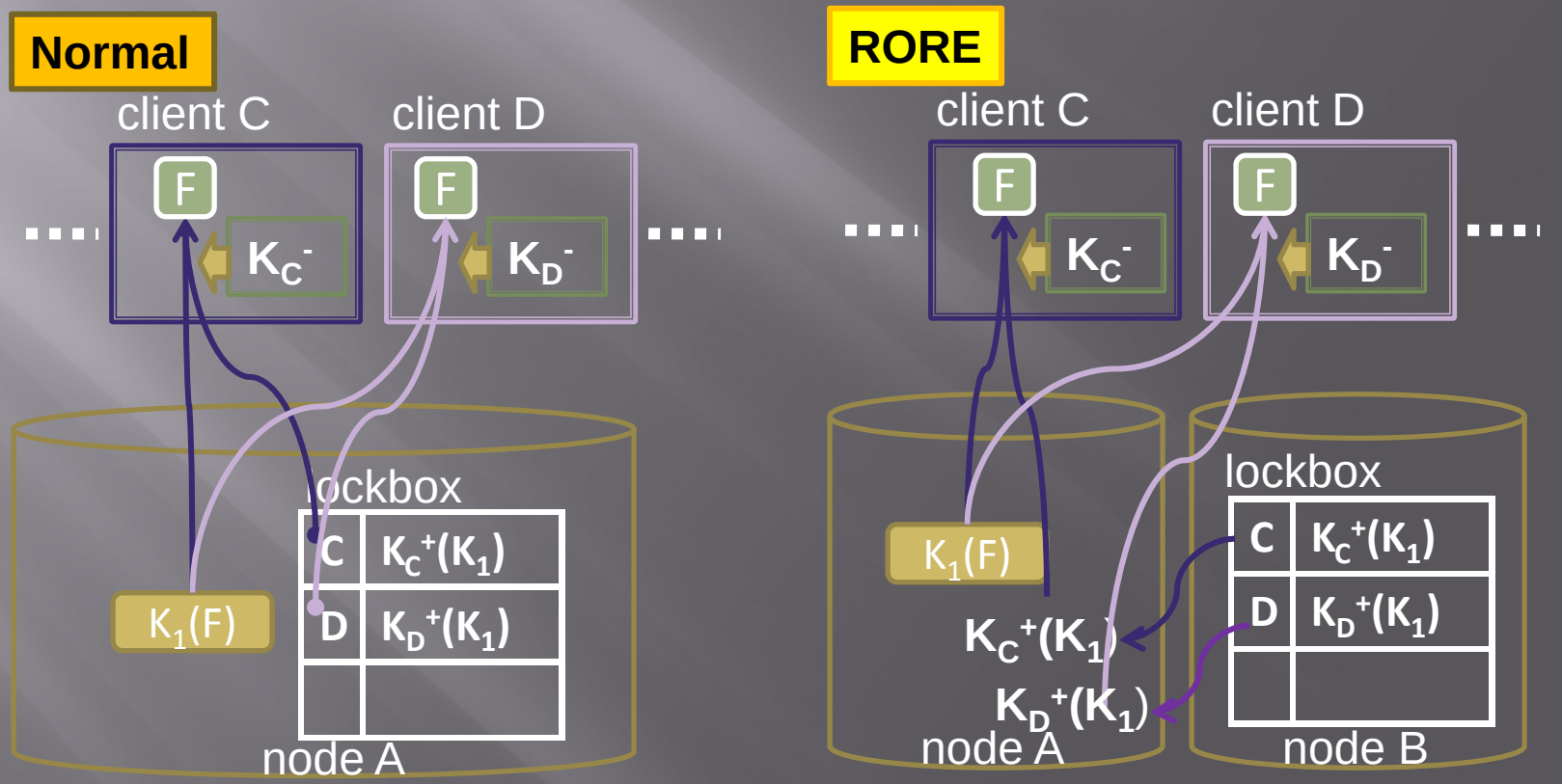
再暗号化の処理時間の比較

- 各環境における再暗号化の処理時間を測定
  - 3台のストレージノードで構成し，各ノードに対し各1台のクライアントから再暗号化要求を実行
  - 1MBのファイルで，3000回の平均を算出
- RORE はNormalと比較して約100ms, 約47%増加
  - RORE ではファイルと鍵を異なるノードに置いている為，古い鍵の獲得や新しい鍵のロックボックスへの設定時に鍵の転送が必要
  - 送受信時の公開鍵暗号による処理が重く，性能へ影響を与えているものと考える



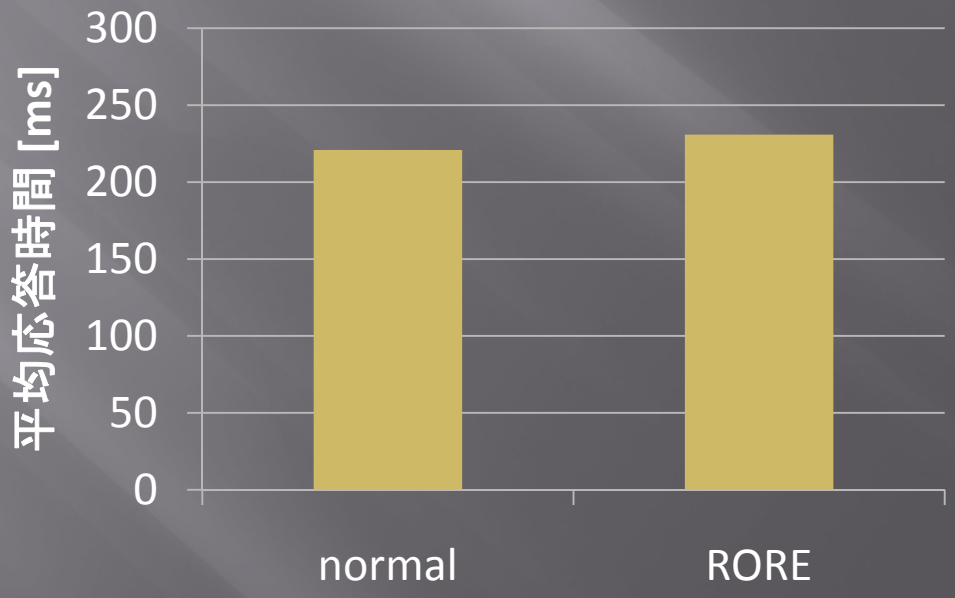
ファイル獲得処理時間の比較

- ファイル獲得（get）の処理の流れ
  - Normal: 格納されたファイルと鍵をクライアントに送信
  - RORE: ファイル格納ノードが鍵を他ノードから受信後，クライアントにファイルと共に送信



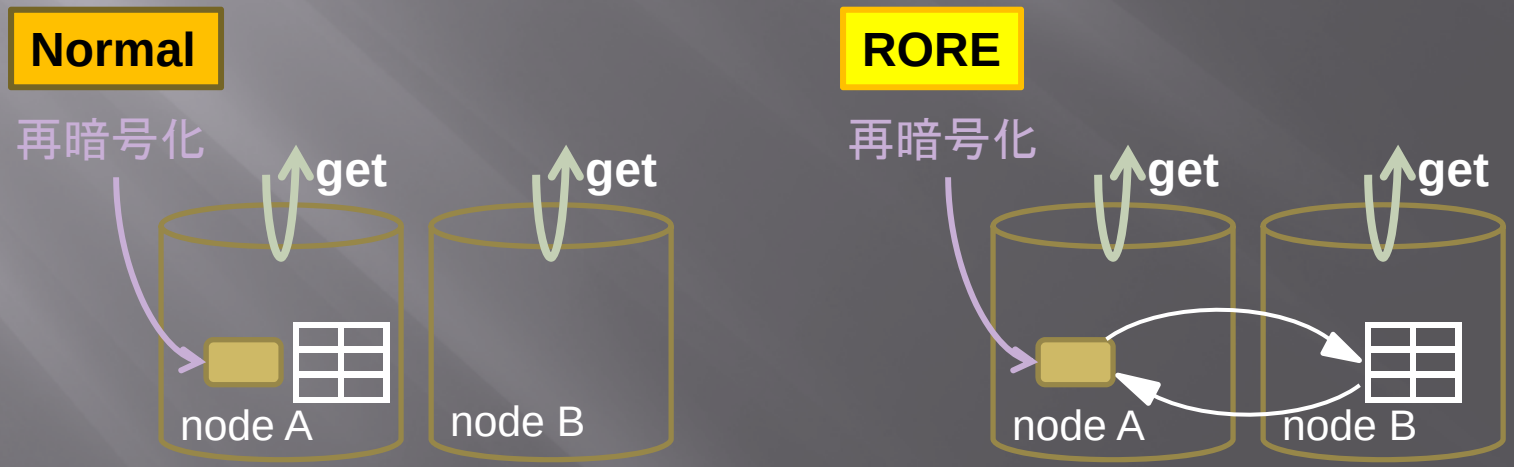
Getの応答時間の比較

- 再暗号化と同様，Getを3000回実行し平均応答時間を算出
- RORE ではストレージノード間で鍵の転送が必要な為，性能劣化が見込まれたが，大きな差はなかった
  - ➔ 鍵のサイズは非常に小さく，その転送による性能への影響は小さい



再暗号化が他アクセスに与える影響

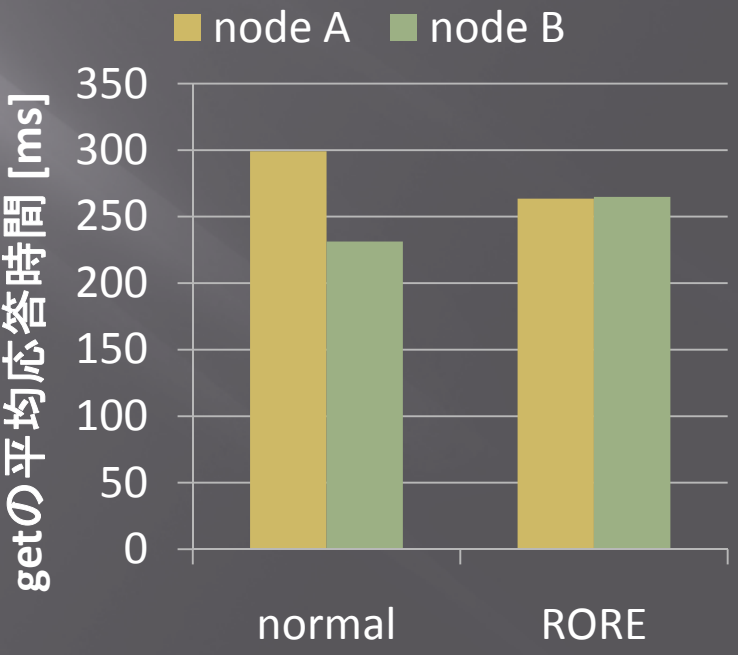
- バックグラウンドでの再暗号化処理が他アクセス(get)に与える影響を測る
- あるノードで再暗号化処理を繰り返している状態で，同時にgetを行い，getの応答時間を測定
  - Normal
    - ノードAに再暗号化対象ファイルとそのロックボックスを格納
    - ノードBはgetのみ
  - RORE
    - ノードAに対象ファイル，Bにそのロックボックスを格納



再暗号化が他アクセスに与える影響

- RORE ではノードA，B共に，getのみの環境（ノードB）と比較して応答時間増化
- normal ではノードAで大きく応答時間が増加

  - RORE では2ノードに再暗号化に関する処理が分散される為，増加量は小さい
  - normal では1ノードに再暗号化の処理が集中するため，そのノードの性能が大きく劣化する



性能面に関する考察

- RORE では再暗号化処理にかかる時間が増加した
  - ノード上の機密性実現のため，鍵をファイルと異なるノードで管理したため
  - 再暗号化処理中のファイルへのアクセスがある場合，処理終了まで待機させられ応答が大きく遅延する可能性
- バックグラウンドでの再暗号化処理の場合，RORE では性能劣化は複数ノードに分散し，1ノード当たりの劣化度合は小さい
  - Normal では1ノードで大きく性能が落ちる可能性
  - QoSの観点では，RORE は突出して性能が落ちるノードがない為優れているといえる
  - 今後，再暗号化はバックグラウンドで実行されるBA-Revへの適用を想定
- セキュアチップ等を使い内部で鍵を管理することにより性能改善の可能性がある

セキュリティ面に関する考察

- 実験の環境では，RORE は攻撃者による1ノード陥落に対し機密性を保証
  - ファイル格納ノードと鍵格納ノードのうち両方が陥落した場合はデータが漏洩
  - normal では1ノード陥落でデータ漏洩の可能性
  - → データ漏洩まで複数ステップかかる分，提案は機密性において優れているといえる
- RORE により，ストレージ上で安全な再暗号化処理を実現することで，クライント側の負担を軽減
  - データ管理コストの大きい，大容量分散ファイルサーバ等に適する

- 専用の鍵サーバやセキュリティソフト組込スマートカード等，堅牢な鍵管理構造を使える場合，さらに高い機密性を実現でき，漏洩の危険性を小さくできる

我々の取り組みのまとめ

- ネットワークストレージの通信路の機密性：BA-Rev の提案
  - Encrypt-On-DiskでのRevocation発生時の性能とセキュリティを両立
    - Active Revocationと同等のセキュリティを提供
    - 実験でGetの性能向上を確認
  - 更新時性能劣化の問題に対しDelayed Writingを適用
    - 性能改善
    - 信頼性の低下は無視可能
- ストレージノード上の機密性：RORE
  - 処理中に平文を生成しない再暗号化手法
  - ネットワークストレージ上での再暗号化に適用
    - クライアント側の負担を軽減
    - 攻撃者による1ノード陥落に対し機密性を保証
  - 実験結果
    - RORE では再暗号化処理時間が増加
      - ノード上の機密性保障の為にファイルと鍵を異なるノードで管理
    - 複数ノードで処理を行う為，1ノード当たりのリソース消費は小さくなる
    - バックグラウンドでの再暗号化で通常の再暗号化よりQoSで優れる

Agenda

- データ工学とその動向
- 関係データベースとセキュリティ
- データマイニングとセキュリティ
- XMLデータベースとセキュリティ
- ストレージとセキュリティ
- ネットワークストレージとセキュリティ
  - 我々の取り組み
- まとめ

まとめ

- データ工学ではセキュリティはホットな話題
  - 様々な面からセキュリティが考えられている
  - 通信内容の保護と格納内容の保護
    - 性能面と安全性
  - 機密保護の対象
    - 関係データベースの属性
    - データマイニングの結果からの推論
    - XMLデータベースの階層構造
    - ストレージの格納オブジェクト
- 我々の取り組みの紹介
  - ネットワークストレージにおける耐故障性とセキュリティの組み合わせ