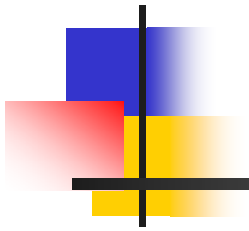


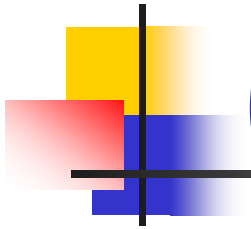
ビルディングオートメーションのIPv6化と安全な自律設定の検討

2004年4月15日

(株)東芝　研究開発センター

井上　淳

(inoue@isl.rdc.toshiba.co.jp)



はじめに

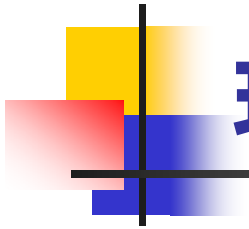
- 本発表は、IPv6普及・高度化推進協議会アプリケーションWG、BA SWGでの議論に基づいている
  - 横河電機、東京大学、東芝の共同研究
- 最新の議論は、以下を参照のこと
  - http://www.v6pc.jp/jp/wg/baSWG/

ビルディングオートメーション (BA) の背景

- ビルを取り巻く新しい要求
  - 環境への配慮
    - COP3：省エネ法により一般ビルでもCO_2削減報告義務付け
  - コスト
    - Life Cycle Cost(LCC)の削減
    - 初期建築費 .vs. 運用管理費
  - ビルの価値
    - 多数の大規模開発により、ビル間競争の激化
    - 顧客 (ビルオーナー、利用者、テナント) に対し、高付加価値をアピール、提供することが重要に

BAシステムの現状

- ビル内はclosedな単一ベンダシステム
  - 縦割りのシステム化
  - コスト構造が不明瞭→競争が働かない
  - 拡張性に問題
- オープンBAへの期待
  - 誰でも（多様なベンダ製品が利用できる）
  - 自由に（データ公開）
  - 簡単に（特別なインタフェース不要）
  - ダウンサイジング化、自律分散化
  - 管理の高度化、低コスト化
- 例：LonWorks(Echelon)、BACnet(ASHRAE)

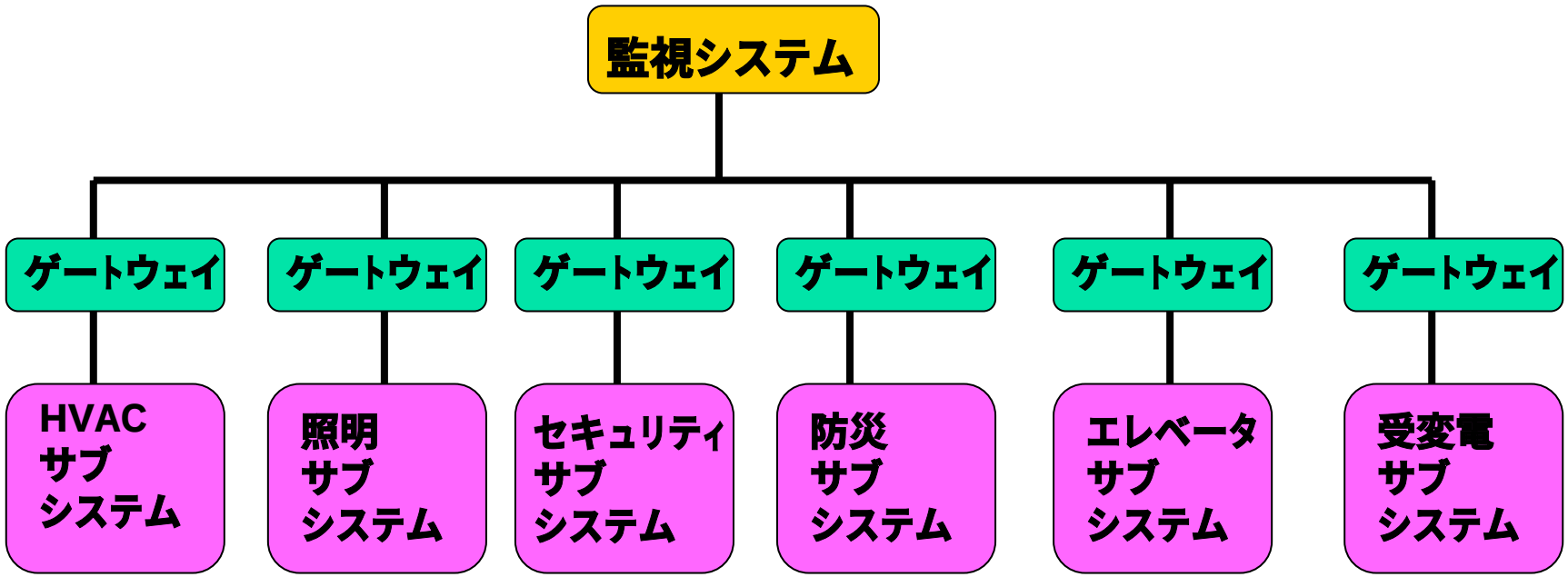


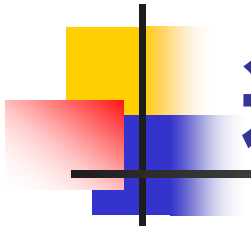
現状のOpen BA (例：LonWorks)の問題点

- プロトコル設計が古い
- チップ (8ビット) の処理能力が限定されている
- 通信バンド幅がない
- チップ毎にEchelonに使用料を支払う必要 (bind)
- 同時利用ノード数の限界
  - 必然的にGatewayを経由する縦割り構造
  - Bootstrapに高い労力
- セキュリティメカニズムが (非常に) 甘い

現状のOpen BAの問題点

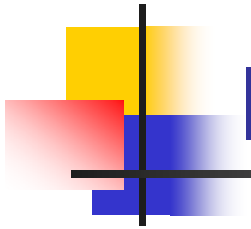
- サブシステム毎にクローズ
- 連携に手間





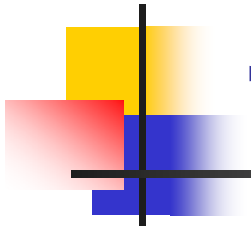
適用事例

- 六本木ヒルズ
  - 54階、述べ床面積380,000平方メートル
  - 監視点数：160,000点
  - フィールド制御
    - LonWorks (空調等)
    - NMAST (照明、動力系)
  - 監視・制御系はTCP/IP
- 松下電工本社ビル
  - 実験的にフィールドまでIPv4を利用 (EMITプロトコル)



IPv6化の意義

- コスト削減
  - 他の通信機器との技術共有可能 (設計、部品)
  - インフラの共有
- 管理系ネットワークと制御系ネットワークの融合
  - オペレーション効率向上
- BAシステム自体の柔軟性向上
- ユーザへの新しいサービス提供
  - ネットワーク仕様をユーザに開放することで利便性向上
- 新規技術、市場の開拓



アプリケーション例：六本木ヒルズ

- R!クリックサービス
  - http://r-click.jp/
  - RFIDと携帯電話を利用した位置利用サービス
  - まだビル自身の機能とは無関係
  - ビル制御ネットワークとの連携
    - 例：入退室管理と空調・照明システムの連動

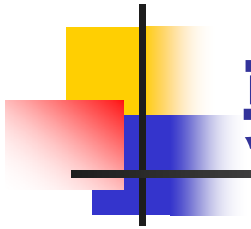
アプリケーション例：その他の構想

- 入退室管理システムと照明・空調の連係
  - 建物全体のエネルギー消費量を削減
- セキュリティシステムとエレベータを連係
  - テナントが全てクローズしたフロアには一般エレベータは止めない→フロア全体の安全性強化
- 設備系以外の端末を協調動作
  - 自分の携帯電話で空調を直接制御
  - 警備会社のシステムとフロアのカメラを接続

  →固定系ノードだけでなく、アドホック・テンポラルなゲストユーザの認証された接続方式が必要

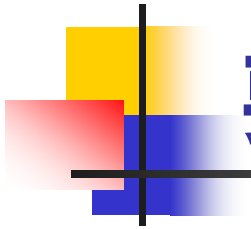
制御ネットワークのIPv6化への課題

- ノード数に対する管理容易性
  - 今までにないスケールのノード数を管理運用
- セキュリティ
  - 一般ユーザへの解放：誤った操作・接続の検出・拒否が必要
  - 仕様が公開→攻撃者のハードルは低い
  - 従来は仕様のcloseさに依存→セキュリティの概念は殆どない
- 個々のノードのコスト
  - 安価なCPU、メモリサイズ



要求条件

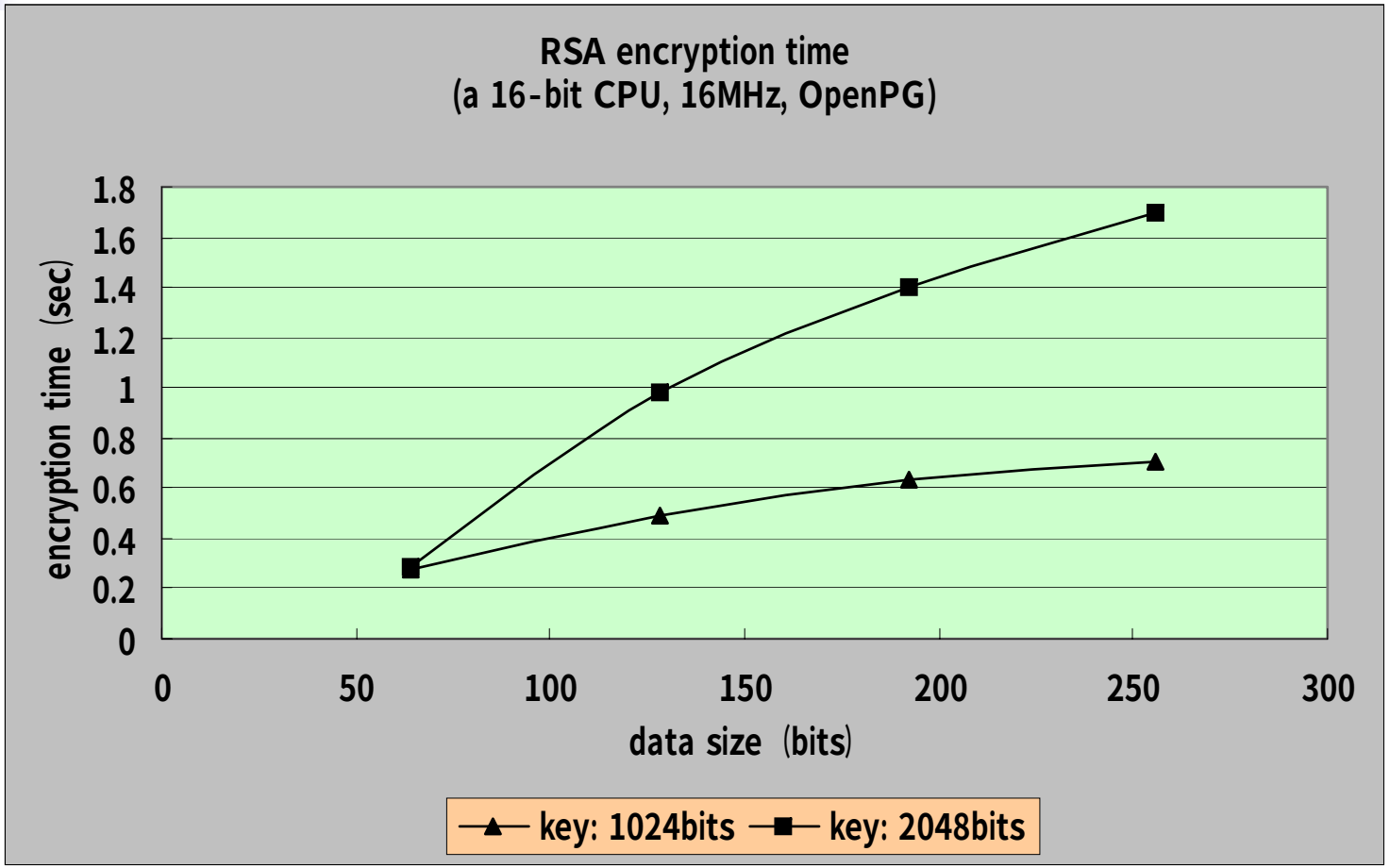
- ノードが自身の設定情報を自律的に取得する
  - 例：「自分はこの蛍光灯をON/OFFする
- 低い管理コスト
  - ノードが自立的に自己設定可能
  - 一括でネットワーク経由で管理できる
  - ノード個別情報は使用環境に非依存としたい
    - ベンダ出荷時に、固有情報を設定可能
  - 個々のノードの起動時間に配慮不要
- ノード数に対するscalability
  - 設定情報が分散管理可能



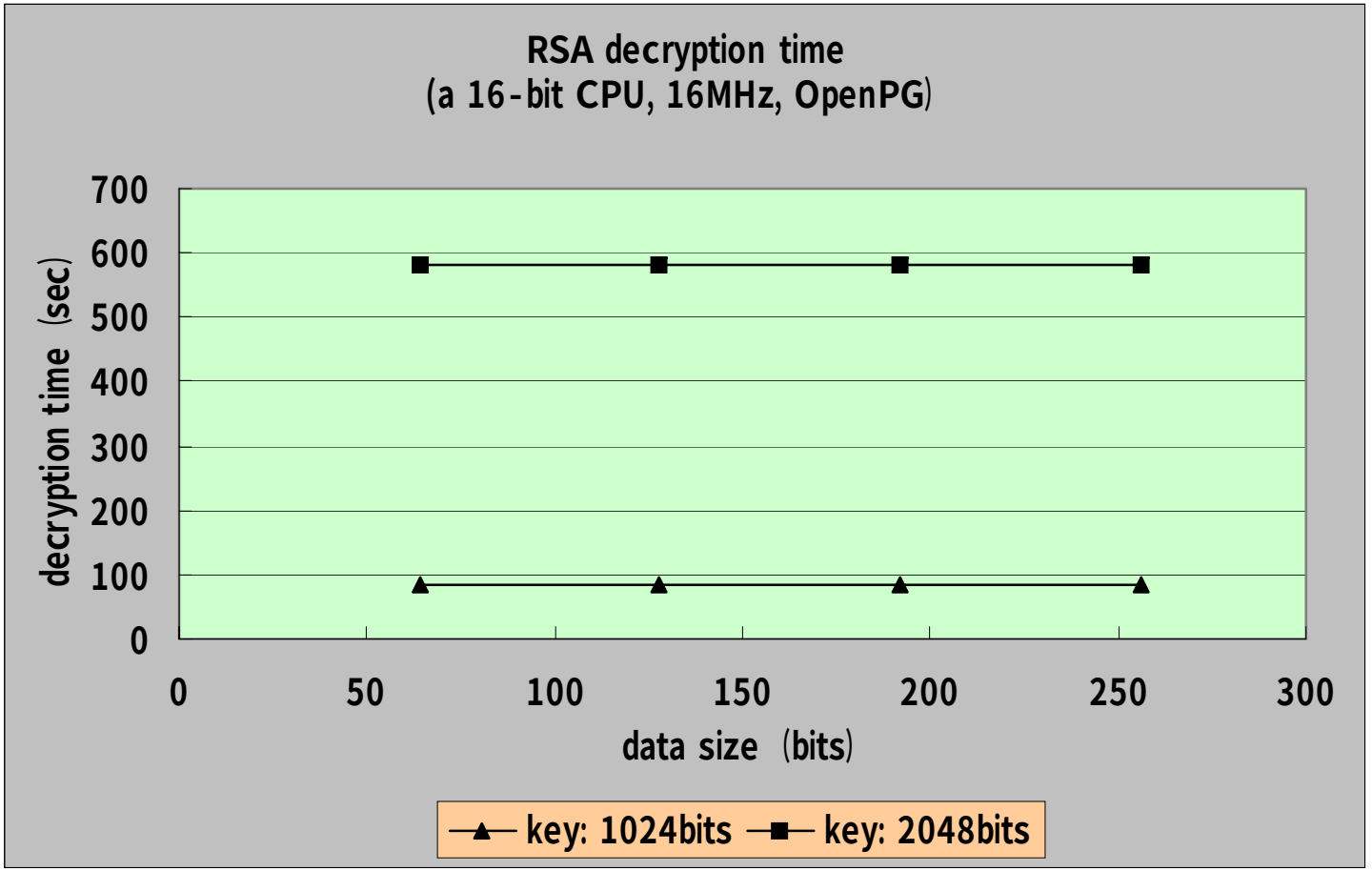
要求条件(cont'd)

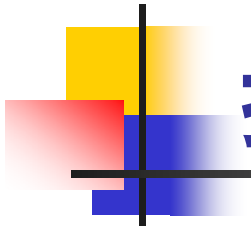
- セキュリティ
  - 盗聴やなりすましに対する防衛
  - 誤った設定情報を与える攻撃に対する防衛
  - 低性能なノードでもセキュリティ配備可能
- 低コストノードでの運用可能性
  - スイッチ、センサーなども対象
  - ノードに強力な計算能力を期待しない
  - 公開鍵暗号系は導入が難しい
    - 多倍長整数演算のコスト大

暗号処理の問題


RSA encryption time
(a 16-bit CPU, 16MHz, OpenPG)
encryption time (sec)
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
data size (bits)
key: 1024bits
key: 2048bits

暗号処理の問題(cont'd)

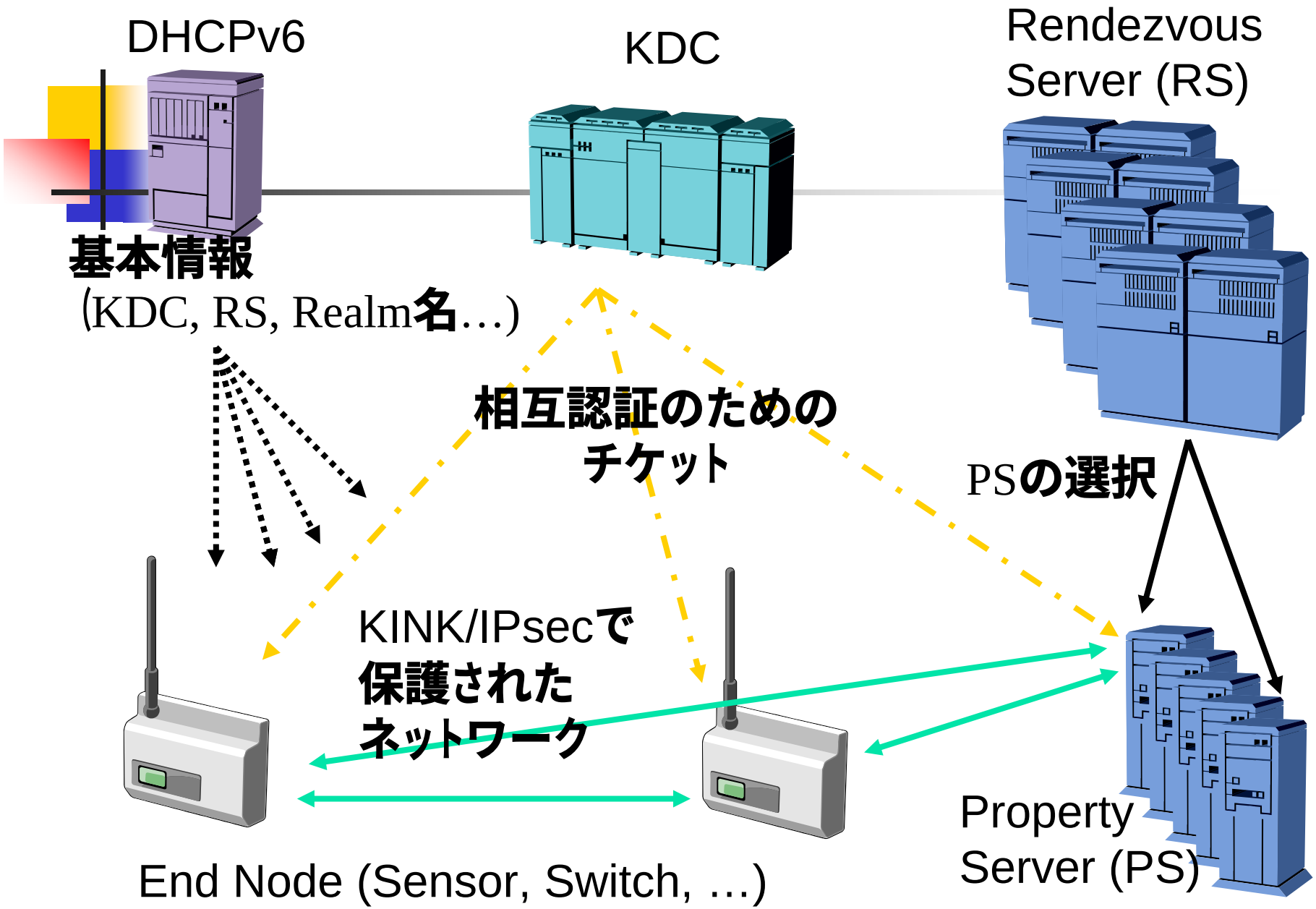


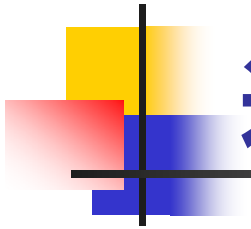


提案方式

- 以下の構成要素から成るKerberosベースのシステム
  - Property Server (PS)
    - 各ノードの設定情報を管理するサーバ
  - Rendezvous Server (RS)
    - あるIDに関連する設定情報をどのPSが受け持つかを管理
  - ノード
    - 固有ID (EUI-64) とKDCとの共有鍵を予め持つ
    - IPsecで通信
  - 鍵管理プロトコル
    - KINK (KerberosベースのIPsec鍵交換プロトコル) を使用

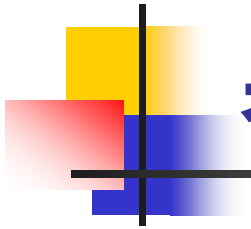
システム構成


DHCPv6
KDC
Rendezvous Server (RS)
基本情報
(KDC, RS, Realm名…)
相互認証のための
チケット
PSの選択
KINK/IPsecで
保護された
ネットワーク
Property Server (PS)
End Node (Sensor, Switch, …)



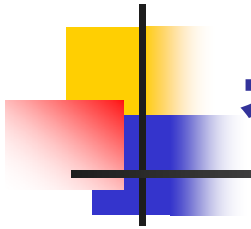
通信手順

1. 初期情報取得
   - DHCPv6でノード非依存なサーバ（KDC,RSなど）アドレス、Realm名を獲得
2. KDC認証
   - TGT獲得
3. PSの発見
   - RSに自身のIDに対応するレコードで問い合わせ
4. PSと鍵交換、設定情報取得
   - KINKで鍵交換を行い、IPsecをかけて安全に情報取得



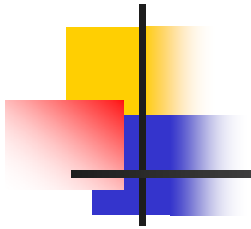
考察

- 管理コスト
  - 個々のノードに設定するのはIDとKDC鍵のみ
  - ノードの配置先に依存しない情報のみ
  - 他の情報は全てPSに集約
    - ネットワーク経由で安全に自律的に設定
- セキュリティ
  - 全ての（PSを含む）ノード間通信はIPsecで守られる
  - 起動後に鍵を更新すれば、デフォルト鍵を廃棄する運用も可能



考察(cont'd)

- Scalability
  - 負荷分散が容易
    - KDCは容易に多重化可能
    - PSもRSへの設定を変更すれば分散化可能
    - RSはDNS同様分散化可能
- ノード性能の制約
  - 鍵交換はIKEでなくKINK
    - 共通鍵暗号のみ (多倍長整数演算不要)



まとめ

- 制御ネットワークのIPv6化
  - 膨大な数のノードが接続
  - 低コスト化、高付加価値化
  - セキュリティの必要性
- 安全で自律的なノード初期設定方式
  - Kerberos+IPsecでセキュリティ提供
  - ノード自身の自律的設定情報発見
    - Property Server + Rendezvous Server
  - 個々のノードへの初期情報設定
    - IDとKDC鍵のみ：配置先に非依存
  - 低コスト・性能ノードでも実現可能
    - 共通鍵暗号のみ