Overview sulla sicurezza nelle reti di sensori

Francesca Bigardi

15 aprile 2012

Rete di sensori

Una rete di sensori è composta da un certo numero di piccoli nodi sensori che sono tra loro densamente distribuiti nell'area di interesse da analizzare o molto vicino ad essa.

- **Obiettivo principale:** raccogliere informazioni e dati dall'ambiente fisico circostante.
- I nodi rilevano i cambiamenti relativi a determinati parametri od eventi e li comunicano agli altri dispositivi.
- Distribuzione dei nodi:
  - La posizione dei nodi non è né predetermina, né organizzata a priori.
  - I nodi sensori possono essere anche utilizzati in ambienti poco accessibili o inacessibili.
  - Richiedono nessuna o poca iterazione umana.

Reti di sensori - Cont.

- Gli algoritmi e i protocolli delle reti di sensori devono avere capacità di auto-organizzazione.
- **Caratteristica importante:** I nodi sensori devono cooperare tra di loro.
- I nodi sensori sono dotati di un processore integrato: eseguono semplici computazioni in modo da trasmettere solo i dati richiesti e parzialmente processati. Non trasmettono i dati grezzi.
- La realizzazione di applicazioni per le reti di sensori richiedono tecniche wireless e di ad hoc networking.
- Non tutti gli algoritmi proposti per le tradizionali reti wireless ad hoc si adattano ai requisiti applicativi e alle caratteristiche delle reti di sensori.

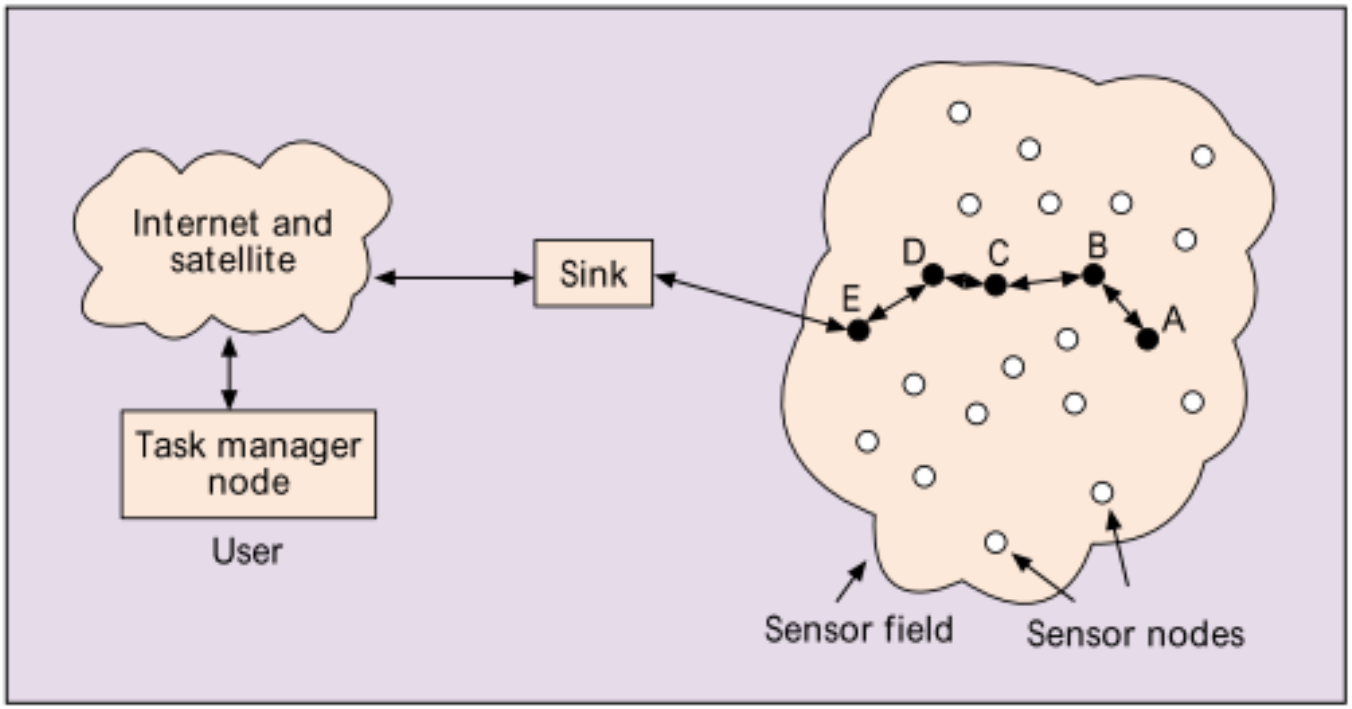
Differeze tra reti di sensori e reti ad hoc

- Il numero di nodi sensori in una rete di sensori può essere di diversi ordini di grandezza più grande del numero di nodi in una rete ad hoc.
- I nodi sensori sono densamente distribuiti ed inclini ai fallimenti
- La topologia di una rete di sensori cambia molto di frequente.
- I nodi sensori hanno risorse limitate in potenza, capacità di calcolo, memoria ed energia.
- Le reti di sensori tipicamente necessitano di relazioni fidate tra i nodi.

Differeze tra reti di sensori e reti ad hoc - Cont.

- I nodi sensori usano diversi paradigmi di comunicazione, mentre la maggior parte delle reti ad hoc si basano su comunicazioni point-to-point. Tra questi paradigmi per le reti di sensori ci sono:
  - **many-to-one**: più nodi sensori inviano i dati ad una base station o ad un punto di aggregazione nella rete.
  - **One-to-many**: un singolo nodo comunica via multicast una query o un'informazione di controllo a parecchi nodi sensori.
  - **Local communication**: per scoprirsi e coordinarsi l'uno con gli altri i nodi vicini inviano dei messaggi. Un nodo può spedire i messaggi a tutti i nodi vicini via broadcast o ad un singolo nodo vicino in modo unicast.
- I nodi sensori possono non avere un'ID globale a causa della grande quantità di overhead e del grande numero di sensori.

Architettura di comunicazione



- Architettura di comunicazione multihop.
- I dati sono instradati verso il sink.

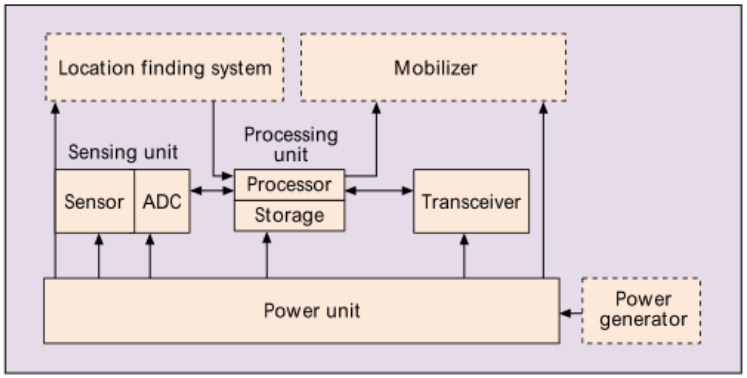
Fattori che influenzano il design di una rete di sensori

- **Tolleranza agli errori**: la rete di sensori deve riuscire a far fronte ai fallimenti dovuti ad esempio a mancanza di energia, danni fisici, interferenze ambientali.
- **Scalabilità**: il numero di sensori e anche la densità di nodi è un paramentro importante da considerare quando si progetta una rete di sensori.
- **Costi di produzione**: il costo delle reti di sensori deve essere più basso di quello che si avrebbe nel sviluppare i tradizionali sensori.
- **Ambiente**: Le reti di sensori di solito operano in aree geografiche remote e non controllate.

Fattori che influenzano il design - Cont. 1

- **Topologia della rete di sensori**: ci sono tre fasi che riguardano la topologia di una rete di sensori:
  - *Fase di pre-distribuzione e di distribuzione*: come i nodi sensori vengono distribuiti nel sensor field.
  - *Fase di post-distribuzione*: come la topologia cambia a causa del cambiamento della posizione dei nodi, della loro raggiungibilità, dell'energia disponibile, dei malfunzionamenti.
  - *Fase di re-distribuzione di nodi addizionali*
- **Mezzo di trasmissione**: nelle reti di sensori multihop, i nodi sono collegati da un mezzo wireless: radio, infrarossi, mezzi ottici.
- **Consumo energetico**: i nodi sensori possono solo essere dotati di limitata energia. La durata di vita di un nodo è strettamente collegata alla durata di vita della sua batteria.

Fattori che influenzano il design - Cont. 2



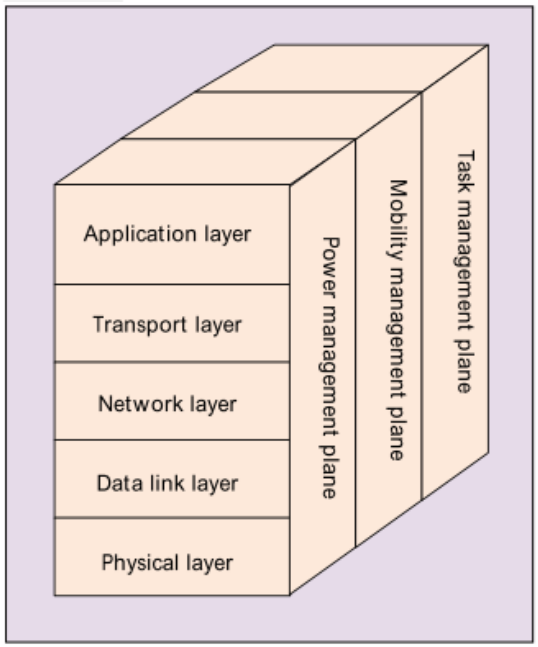
- **Vincoli hardware**. Un nodo è composto da:
  - Un'unità di rilevamento, composta da: sensori e convertitori da analogico a digitale (ACD);
  - Un'unita di elaborazione, associata con una piccola unità di memorizzazione. Permette ai nodi di colloborare con gli altri nodi al fine di svolgere le proprie mansioni.
  - Un'unità di trasmissione che connette il nodo alla rete.
  - Un'unità di consumo.
  - Componenti addizionali in base al compito che devono svolgere.

Caratteristiche nodo sensore

- Un nodo sensore deve:
  - consumare poca energia;
  - operare con altà densità volumetrica;
  - avere bassi costi di produzione;
  - essere superfluo e autonomo;
  - operare senza sorveglianza;
  - sapersi adattare all'ambiente.

Protocol stack

- **Physical layer**: esegue semplici ma robuste tecniche di modulazione, trasmissione e ricezione.
- **Data link layer**: deve minimizzare le collisioni con i vicini durante le trasmissioni broadcasts ed essere consapevole dell'energia consumata.



Protocol stack - Cont.

- **Network layer**: si occupato del routing dei dati.
- **Trasport layer**: aiuta a mantenere il flusso dei dati, se richiesto dall'applicazione delle reti di sensori.
- **Applicaiton layer**: in base ai dati raccolti, differenti applicazioni possono essere costruite e usate nel livello applicazione
- **Power management plane**: monitora il consumo energetico.
- **Mobility managment plane**: monitora il movimento dei sensori.
- **Task management plane**: monitora la distribuzione dei task tra i nodi sensori.

Esempi di ambiti applicativi

- Monitoraggio di ambienti industriali: come strumentazioni, livello di inquinamento, allarme incendi, integrità stutturale degli edifici.
- Monitoraggio e controllo del clima.
- Gestione del consumo energetico / Distribuzione dell'energia.
- Ambiti sanitari, per monitorare pazienti e assistere ai pazienti disabili e biomedici.
- Ambiti commerciali: gestione degli inventari, monitoraggio della qualità dei prodotti.
- Ambito militare: sorveglianza dei campi di battaglia
- Monitoraggio di aree colpite da disastri.

Esempi di ambiti applicativi - Cont.

- Domotica
- Monitoraggio ambientale
- Controllo all'interno di veicoli
- Geolocalizzazione e tracking
- HCI (riconoscimento dei gesti, tracking)
- I sensori possono monitorare:
  - temperatura, pressione, umidità, movimento dei veicoli, livelli di rumore, condizioni di luce, presenza o assenza di certe sostanze, oggetti...

Standard IEEE 802.15.4

- Standard che definisce il livello fisico e MAC di low-rate wireless personal area networks (LR-WPAN).
- È gestito dal gruppo IEEE 802.15.
- ZigBee si basa su questo standard.

ZigBee

Suite di protocolli di comunicazione di alto livello per dispositivi compatti, a basso consumo e a bassa velocità di trasmissione.

Security properties

- **Confidenzialità**: agenti non autorizzati non possono leggere le informazioni. Pressupone che ci sia una security policy che dica chi o cosa può accedere ai dati.
- **Integrità**: i dati non possono essere alterati da agenti non autorizzati. Anche in questo caso pressupono una politica di sicurezza che affermi chi o cosa possa alterare i dati.
- **Disponibilità**: l'agente può accedere ai dati/servizi ogni volta che desidera. Assicurare la disponibilità significa prevenire gli attacchi di denial of service (DoS).

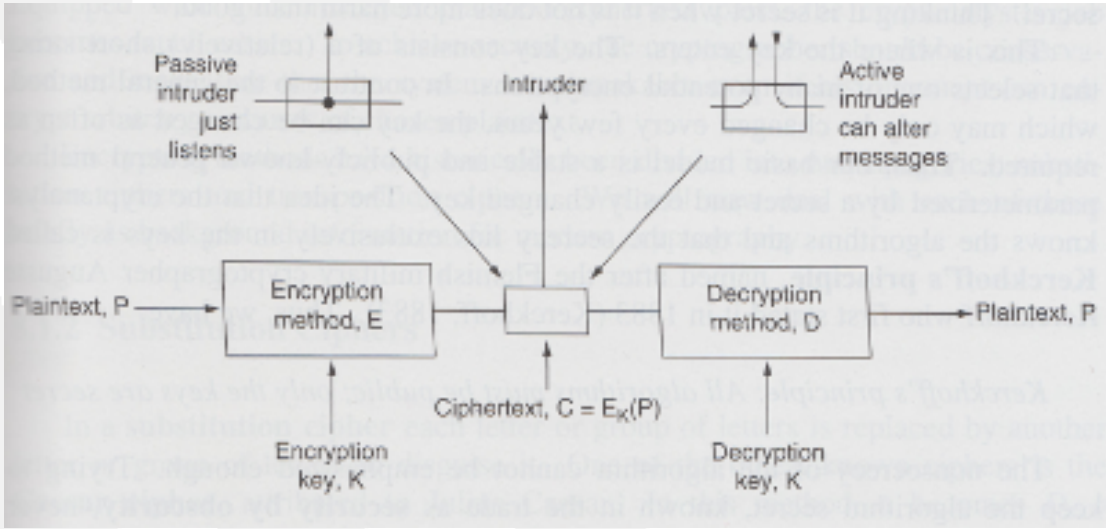
Security properties - Cont.

- **Responsabilità**: è sempre possibile risalire all'agente responsabile di una certa azione.
- **Non-ripudiazione**: è una forma più forte di responsabilità, l'agente non può negare le azioni compiute.
- **Autenticazione**: gli agenti o l'origine dei dati possono essere identificati in modo preciso. Attraverso l'autenticazione si verifica l'indentià di un agente in modo preciso.

Algoritmi a chiave simmetrica

Algoritmi a chiave simmetrica

La stessa chiave viene usata sia per la codifica che la decodifica.



Algoritmi a chiave simmetrica - Cont.

- **DES** - Data Encryption Standard: usa blocchi da 64 bit con chiavi da 56 bit.
- **Triple DES**: Usa due chiavi da 56 bit ed esegue tripla cifratura. Usa al suo interno DES.
- **AES** - Advanced Encryption Standard: AES ha due varianti: una con blocchi da 128 bit e chiavi da 128 bit e una con blocchi da 128 bit e chiavi da 256 bit.

Message authentication code (MAC)

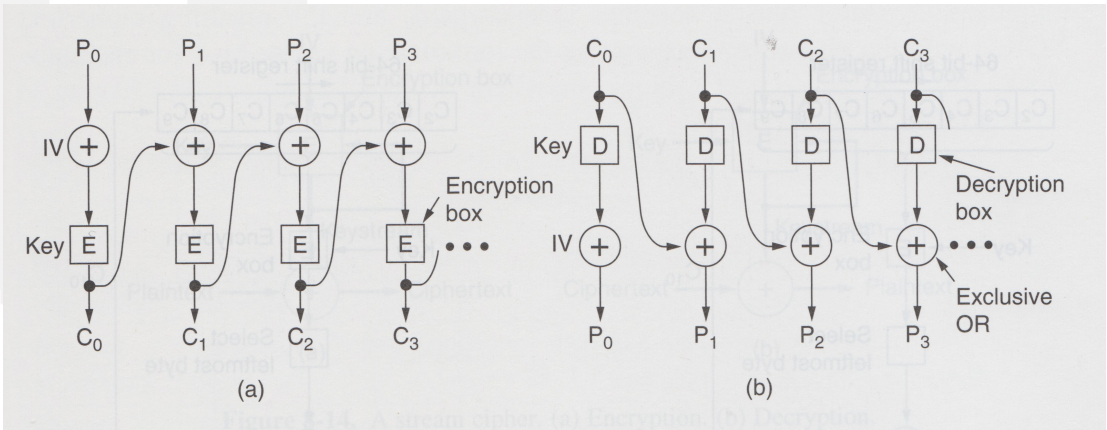
- Soluzione comune per raggiugere autenticazione e integrità di messaggio
- Si può vedere un MAC come un checksum sicuro dal punto di vista crittografico.
- Per computare un MAC mittente e ricevente devono condividere una chiave segreta, che viene usata come input nel calcolo.
- Il mittente computa un MAC su un pacchetto con la chiave segreta e pone il MAC calcolato nel pacchetto.
- Il ricevente ricomputa il MAC e lo confronta con quello ricevuto.
- Accettando o meno il pacchetto in base al confronto.

Initialization vector (IV)

- È un input a dimensione fissa per una primitiva crittografica, deve essere tipicamente casuale o pseudo-casuale.
- Aggiunge una variazione al processo crittografico quando c'è una piccola variazione nell'insieme di messaggi.
- IV vengono usati in fasi di decodifica, sono quindi inviati in chiaro e inclusi nello stesso paccheto dei dati cifrati.

Chipher Modes

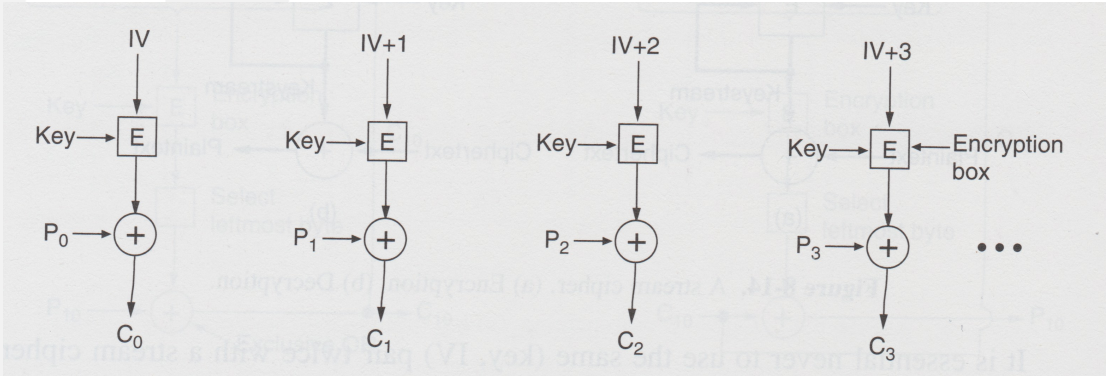
- **CBC Mode** - Chipher Block Chaining Mode: ogni blocco del plaintex viene messo in XOR con il precedente blocco cifrato prima di essere codificato.



  - Il primo blocco è messo in XOR con un **IV** (Initialization Vector) che è trasmesso (in chiaro) con il ciphertext.
  - Lo stesso blocco nel plaintext non risulterà più nello stesso blocco del ciphertext.

Chipher Modes - Cont. 1

- **Counter Mode**:



  - Il plaintext non viene cifrato direttamente.
  - Si cifra IV più una costante e il ciphertext risultante viene messo in XOR con il plaintext.
  - Aumentando l'IV di 1 per ogni nuovo blocco, si può decifrare un blocco ovunque nel file senza aver prima decifrato tutti i suoi predecessori.
  - Le chiavi e l'IV dovrebbero essere scelti indipendentemente e casualmente.

Chipher Modes - Cont. 2

- **CCM mode**: combina Counter mode con CBC-MAC mode.
- **CBC-MAC mode**: è una tecnica per costruire un message authentication code da un block cipher. Il messaggio è cifrato con un qualche algoritmo di block cipher in CBC mode.

Algoritmi a chiave pubblica

- La distribuzione di chiavi è un punto critico nei criptosistemi.

Algoritmi a chiave pubblica

Le chiavi di codifica e decodifica sono differenti, e la chiave di decodifica non si può derivare dalla chiave di cifratura.

- $D(E(P)) = P$
- È estramemente difficile dedurre D da E
- E non può essere rotto da un chosen plaintext attack.
- La chiave e l'algoritmo di cifratura sono pubblici. Si parla di chiave pubblica.
- La chiave di decodifica è privata. Si parla di chiave privata.
- **RSA**: richiede chiavi di almeno 1024 bits per avere una buona sicurezza. È troppo lento per cifrare grandi quantità di dati. È usato per la distribuzione di chiavi.

Sicurezza

- Le WSN come internet o altre reti ad hoc possono essere soggette ad attività malevoli.
- La sicurezza è un requisito da cui non si può prescindere.
- I servizi di sicurezza devono essere:
  - efficienti dal punto di vista energetico;
  - scalabili;
  - forti;
- I servizi di sicurezza devono includere:
  - confidenzialità;
  - integrità;
  - autenticazione;

  sia dei dati rilevati dal sensore che delle informazioni di routing.
- Sebbene significativi progressi siano stati fatti nelle WSN, la sicurezza è ancora un campo di ricerca aperto, soprattutto per quanto riguarda confidenzialità e autenticazione.

Possibili attacchi

- Gli attacchi possono riguardare qualsiasi layer del protocol stack.
- Possibili attacchi:
  - cattura del nodo;
  - manomissione fisica del nodo;
  - spoofing;
  - sniffing;
  - denial of service (DoS).

Categorizzazione delle minacce

- Le minacce si possono classificare in:
  - attive:
    - masquerade;
    - replay;
    - modifica dei messaggi;
    - denial of service;
  - passive:
    - analisi del traffico;
    - rilascio del contenuto del messaggio;
  - Interne:
    - Nodi interni compromessi che eseguono codice malevolo.
    - Avversari che hanno rubato il materiale di codifica, il codice e i dati dal nodo legittimo.
  - Esterne: l'attaccante non ha accesso speciale alla rete di sensori.

Attaccanti

- Gli attaccanti si possono distingure in:
  - *mote-class attackers*: l'attaccante ha accesso ha pochi nodi sensori con capacità simili agli altri nodi.
  - *laptop-class attackers*: l'attaccante può avere accesso a dispositivi più potenti, come laptop o simili. Un attaccante in questa classe ha più possibilità di attacco di un semplice nodo.

Uso di algoritmi adatti alle normali reti di computer

- **Domanda:** Si possono usare le tecniche di sicurezza applicate nelle reti dei normali computer?
- **Risposta**: No. Le caratteristiche delle WSN fanno sì che i recenti algoritmi sicuri usati nelle workstations più potenti non siano applicabili.

Caratteristiche WSN incidenti sulla sicurezza

- Le WSN hanno risorse limitate, rispetto ad un normale computer:
  - Limitazioni computazionali
  - Limitazione di memoria
  - Limitazioni di batteria/energia.
  - Limitazioni di comunicazione.
  - Pacchetti di piccole dimensioni (30 byte).
- Distribuzione dei nodi su larga scala.

- Esempio di nodo sensore SmartDust sviluppato da UC Berkeley:
  - **CPU**: 8-bit, 4MHz
  - **Memoria**: 8 Kbyte di instruzioni flash, 512 bytes di RAM, 512 bytes EEPROM
  - **Comunicazione**: radio a 916 MHz
  - **Larghezza di banda**: 10 Kbps
  - **Sistema operativo**: TinyOS
  - **Spazio per codice SO**: 3500 bytes
  - **Spazio disponibile per codice**: 4500 bytes

Uso degli schemi di cifratura

- I protocolli di sicurezza inoltre tendono ad essere conservativi nel garantire la sicurezza, aggiungendo 16-32 byte di overhead.
- L'applicazione degli schemi di cifratura richiede quindi la trasmissione di bit extra, che comportano:
  - elaborazioni extra;
  - memoria aggiuntiva;
  - consumo di batteria extra;
- Inoltre si può avere un aumento di:
  - ritardo;
  - jitter;
  - perdita dei pacchetti

Esempio uso di RSA

- La memoria di un tipico nodo sensore non è grande abbastanza per mantenere le variabili necessarie alla sua implementazione;
- Il tempo di computazione è elevato;
- Il nodo può consumare tutta la batteria a disposizione con una singola computazione;

Sfide

1. Minimizzare il consumo di risorse, massimizzando la sicurezza.
   - L'ordine in cui prendere in considerazione le limitate risorse è questo:
     1. energia;
     2. memoria;
     3. potenza di calcolo;
     4. larghezza di banda della comunicazione;
     5. range di comunicazione;
2. Le funzionalità e i vincoli hardware dei nodi sensori influenzano il tipo di meccanismo di sicurezza che un nodo può ospitare.
   - L'energia aggiuntiva usata per proteggere i messaggi deve essere poca.
   - L'uso maggiore di energia sia ha nel processo di key establishment.

Problemi principali

- **Gestione delle chiavi**.
- **Routing sicuro**.
- **Prevenzione DoS**.
- **Energia**
- **Autenticazione broadcast**

Gestione delle chiavi

- Le chiavi sono necessarie per fornire confidenzialità, integrità e autenticazione.
- Scopo della gestione delle chiavi:
  - inizializzare gli utenti del sistema all'interno di un dominio.
  - Generare, distribuire e installare il materiale di codifica.
  - Controllare l'uso del materiale di codifica.
  - Memorizzare, eseguire il backup e il ripristino del materiale di codifica.
- La gestione delle chiavi è un problema non risolto nelle WSN a causa:
  - della natura ad hoc della rete: la topologia della rete non è conosciuta a priori;
  - della connettività intermittente;
  - delle risorse limitate come energia e range di trasmissione;

Gestione delle chiavi - Cont.

- Tradizionalmente la gestione delle chiavi si basa su un'entità fidata chiamata Certificate Authority (CA).
- CA emette certificati di chiave pubblica ad ogni nodo.
- È pericoloso avere un servizio di gestione delle chiavi basato su una singola CA: è un punto vulnerabile della rete.
- Se la CA è compromessa l'intera rete di sensori lo è.
- Il problema di gestione delle chiavi nelle WSN può essere decomposto in:
  - Key pre-distribution;
  - Neighbor discovery;
  - End-to-end path-key establishment;
  - Isolating aberrant nodes;
  - Re-keying;
  - Key-establishment latency;

Gestione delle chiavi - Key Pre-distribution

- Le chiavi sono installate in tutti i sensori in una fase precedente al loro dispiegamento.
- Ad oggi è l'unica opzione pratica possibile per le rete in cui la topologia è sconosciuta a priori. Altri schemi esistono come:
  - trusted server schemes;
  - self enforcing schemes che si basa su crittografia assimetrica;

  ma non sono adatti alle reti di sensori.
- Possibili problemi sono:
  - caricare l'insieme delle chiavi (key ring) nella memoria limitata di ogni sensore.
  - Salvare l'id di chiave di una key ring.
  - Associare l'id di un sensore con il nodo controllore fidato.

Gestione delle chiavi - Schemi di key sharing

- Ci sono tre semplici modelli di keying:
  - **Network keying**: Le chiavi network sono usate per cifrare i dati e per l'autenticazione di tutta la rete.
    - Usa poche risorse.
    - Facile da gestire.
    - Eccellente in termini di scalabilità e flessibilità.
    - Non cambia quando si aggiungono nuovi nodi.
    - Permette la collaborazione dei nodi.
    - Problemi di robustezza: se un nodo è compromesso, l'intera chiave di rete viene rivelata.

Gestione delle chiavi - Schemi di key sharing Cont. 1

- **Pairwise keying**: le pairwise key sono condivise tra due nodi.
  - Robusta: la compromissione di un nodo non compromette gli altri nodi.
  - Non è scalabile: i costi di memorizzazione crescono con la dimensione della rete.
  - Difficile aggiungere nuovi nodi.
  - Consuma molte risorse in confronto alla network keying.
  - Non sempre in fase di distribuzione si conoscono i nodi vicini di un certo nodo.

Gestione delle chiavi - Schemi di key sharing Cont. 2

- **Group keying**:
  - Combina le caratteristiche di entrambi gli schemi.
  - Condivide una chiave simile alla network keying con un gruppo di nodi.
  - Ogni gruppo ha una chiave differente.
  - La compromissione di un nodo, comporta solo la compromissione del gruppo.
  - Il numero di chiavi aumenta con il numero di gruppi.
  - Difficile da impostare e dipende dall'applicazione.

Gestione delle chiavi - Neighbor-discovery

- Ogni nodo deve scoprire quali sono i suoi vicini, cioè quali nodi sono nel suo raggio di comunicazione.
- Ogni nodo condivide una chiave con questi nodi vicini.
- Questa fase di discovery permette di stabilire la topologia della rete di sensori.
- Due nodi sono tra loro collegati se condividono una chiave.
- Buoni schemi di Neighbor discovery non offrono all'attaccante la possibilità di scoprire le chiavi condivise, che può quindi solo fare analisi del traffico.

Gestione delle chiavi - End-to-end path-key establishment

- A due nodi, che non condividono una chiave ma sono collegati da più hops, deve essere assegnata una path key per una sicura comunicazione end-to-end.

Gestione delle chiavi - Isolating aberrant nodes

- Un nodo anomalo è un nodo che non funziona come dovrebbe.
- Identificare e isolare i nodi anomali che agiscono da nodi intermedi è importante per il funzionamento della rete.
- Un nodo smette di funzionare perché:
  - Ha finito la sua energia.
  - È stato danneggiato da un attaccante.
  - Dipende da un nodo intermedio compromesso e quindi è stato deliberatamente bloccato. Il nodo compromesso può:
    - modificare i dati prima di inoltrarli.
    - comunicare informazioni fittizie alla base station.

Gestione delle chiavi - Re-keying

- Avviene quando le chiavi scadono.
- Le nuove chiavi devono essere generate in modo efficiente per il consumo energetico.
- La durata del re-keying dipende dal livello di sicurezza che si vuole raggiungere.

Gestione delle chiavi - Re-establishment latecy

- La latenza è un significativo impedimento durante l'inizializzazione della rete sicura.
- La maggior latenza è dovuta alle comunicazioni piuttosto che alla computazione.

Gestione delle chiavi - Soluzioni

- **Hybrid key-based protocols**: Un mix di protocolli basati su chiave pubblica che sono più efficienti dal punto di vista energetico rispetto ad un singolo protocollo.
- **Threshold cryptography**:
  - Usa uno schema a soglia (k, n) per distribuire i servizi delle CA ad un insieme di nodi specializzati.
  - Soluzione adatta a reti di sensori pianificate e a lunga durata.
  - Problemi nelle WSN:
    - Alcuni nodi possono smettere di funzionare poiché terminano la loro energia.
    - Si basa su cifratura a chiave pubblica.

Gestione delle chiavi - Soluzioni Cont.

- **Pebblenets**
  - Si basa su cifratura simmetrica.
  - Fornisce group authentication, integrità del messaggio e confidenzialità.
  - È adatto per reti pianificate, distribute e a lunga durata.
  - Problemi:
    - Se un nodo viene compromesso, la forward-secrecy viene rotta.
    - Un server di gestione delle chiavi (nodo specializzato) deve memorizzare le proprie key pair, ma anche le chiavi di tutti i nodi nella rete.
    - Overhead nel firmare e verificare i messaggi di routing sia in termini di computazione che di comunicazione
- **Protocollo LEAP**

Routing sicuro

- Ci sono due tipi di minacce per il protocollo di routing:
  - *Minacce esterne* che possono causare ritrasmissioni e routing inefficace. Le minacce includono:
    - iniezione di informazioni di routing errate;
    - replay di vecchie informazioni di routing;
    - distorsione di informazioni di routing.
  - *Minacce interne*: nodi compromessi.
    - Invio di informazioni di routing malevole agli altri nodi.
    - Difficili da scoprire.
- Molti protocolli di routing non sono stati progettati tenendo conto dei problemi di sicurezza.

Problemi per il secure routing

- **Un meccanismo di autenticazione** che comporti poca computazione e basso overhead di comunicazione.
- **Secure route discovery**:
  - Un meccanismo di route discovery dovrebbe:
    - permettere al target di verificare l'autenticità del Route Requestor.
    - autenticare i dati dei messaggi *route request* e *route reply* attraverso l'uso di chiavi di autenticazione per ogni direzione.
  - I nodi compromessi devono essere evitati in fase di Route Discovery.

Problemi per il secure routing - Cont.

- **Route maintenance**:
  - Deve garantire i route error messages.
  - Deve prevenire che nodi non autorizzati inviino questi messaggi.
- **Defending from routing misbehavior**: bisogna determinare se i nodi intermedi hanno inoltrato effettivamente i messaggi.
- **Defending from flooding attack**:
  - Un attaccante può degradare le performance di un protocollo di routing continuando a richiedere la route discovery.
  - Un attaccante invia i pacchetti di Route Request che innondano la rete.
  - È necessario un meccanismo che autentichi le route requests, affinché i nodi possano filtrare le troppe o le falsificate route request.

Routing sicuro - Soluzioni

- **SPINS** - (Security Protocols for Sensor Networks): comprende al suo interno
  - **SNEP** che fornisce confidenzialità, two-party data authentication, integrità e freshness.
  - μ**TESLA** fornsice l'autenticazione broadcast dei dati.
- **Ariadne**:
  - Ogni nodo deve essere in grado di generare una one-way key chain.
  - Un nodo a causa dei limiti di memoria non non può generare una key chain troppo lunga.
  - Un nodo non può spendere troppo tempo nel generare le chiavi.
- **INSENS**: anche se un nodo malevolo può compromettere un piccolo numero di nodi nella sua vicinanza, non può causare un danno all'intera rete.

Prevenzione DoS

- L'obiettivo di un attacco DoS è quello di rendere una macchina inaccessibile agli utenti legittimi.
- Un attacco DoS è un qualsiasi evento che diminuisce o elimina le capacità della rete di eseguire le proprie funzionalità. Oltre ad attacchi effettivi possono causare denial of service:
  - fallimenti hardware;
  - bug software;
  - eusarimento delle risorse;
  - condizioni ambientali;
- Anche l'attaccante può sfruttare questi problemi per eseguire attacchi di denial of service.

Prevenzione DoS - Cont.

- Due tipi di attacchi DoS:
  - attacchi passivi: nodi egoisti che usano la rete ma non cooperano al fine di salvare la durata della loro batteria.
  - attacchi attivi: nodi malevoli che danneggiano gli altri nodi causando l'interruzione della rete stessa.
- Gli attacchi DoS si possono verificare in più livelli del protocol stack.

Prevenzione DoS - Soluzioni

- Meccanismi di autenticazione. L'autenticazione pone però seri problemi nelle reti di sensori.
- Schema watchdog: tenta di identificare e bypassare i nodi che hanno cattivi comportamenti usando watchdog e pathrater.
- Schema di rating: i vicini di ogni singolo nodo collaborano al fine di valutare i nodi.
- Moneta virtuale: permette di evitare attacchi passivi, concettualizza i motivi per cui i nodi non devono essere egoisti.
- Route dos prevention: tenta di prevenire gli attacchi DoS favorendo la cooperazione tra più nodi.

Energia

- Molta energia è consumata nelle comunicazioni piuttosto che nelle computazioni.
- Per salvare l'overhead di comunicazione è necessario salvare i dati in qualche nodo sensore vicino
- Ogni nodo sensore quindi dovrebbe condividere una chiave globale con la stazione base (sink) ma anche condividere link kyes con i sensori aggregatori.
- Lo schema di sicurezza dovrebbe essere energy-efficient.

Autenticazione broadcast

- Un attaccante non dovrebbe essere in grado di dichiarare se stesso come base station e mandare via broadcast falsi comandi ai nodi sensori.
- Si può usare SPINS, che al suo interno contiene $\mu TESLA$.

Attacchi

- Gli attacchi nelle WSN sono simili a quelli che si osservano in una semplice rete di computer o Internet, ma non sono limitati ad essi.
- Gli attacchi sono possibili verso tutti i livelli del protocol stack.

Attacchi sui vari livelli

- **Livello fisico**:
  - manomissione. Possibili soluzioni sono schemi di gestione delle chiavi efficienti, tamper-resistant.
  - Signal jamming: trasmissione di un segnale radio che distrugge le comunicazioni decrementanto il rapporto segnale radio. È un attacco di tipo DoS.
  - **Possibili soluzioni al signal jamming**: spread-spectrum e frequency-hopping.
  - Frequency hopping è una combinazione dinamica di parametri come:
    - hopping set (frequenze disponibili per hopping);
    - dwell time (intervallo di tempo per hop);
    - hopping pattern (la sequenza in cui sono usate le frequenze dall'hopping set disponibile).
  - Frequency hopping costa poco in termini di memoria, elaborazione ed energia.
  - La sequenza di hopping dovrebbe essere modificata in un tempo inferiore a quello richiesto per scoprirla.
  - Mittente e ricevente devono quindi mantenere un clock sincronizzato.

Attacchi sui vari livelli - Cont. 1

- **Livello data link**:
  - È vulnerabile, in quanto i dati sono trasmessi attraverso un mezzo insicuro.
  - È suscettibile ad attacchi su autenticazione, integrità e confidenzialità dei dati che devono essere instradati.
  - Possibili attacchi sono:
    - Esaurimento risorse. Si può risolvere limitando il rate di trasmissione.
    - Collisioni. Una possibile difesa è l'error correcting code.

Attacchi sui vari livelli - Cont. 2

- **Livello di rete**: a livello di rete sono possibili svariati attacchi:
  - Spoofing, altered or replaying routing information
  - Selective forwarding or black holes
  - Sink holes
  - Sybil attacks
  - Wormholes
  - Flooding
  - Attacks against privacy

Livello di rete - Spoofing, altered or replaying routing information 1

- Attraverso lo spoofing, l'alterazione o la replica di informazioni di routing il traffico di rete può essere corrotto.
  - Si possono creare routing loops;
  - generare falsi messaggi di errore;
  - partizionare la rete;
  - incrementare la latenza end-to-end;
  - aumentare o diminuire il traffico di rete;
  - aumentare o diminuire i percorsi originari;
  - ...

Livello di rete - Spoofing, altered or replaying routing information 2

- Questi attacchi possono portare ad una perdita dei pacchetti che a sua volta causa continue ristrasmissioni.
- Continue ritrasmissioni riducono la durata di vita di un nodo.
- In generale tali attacchi possono provocare una diminuizione delle performance della rete.
- **Soluzioni**:
  - Lo spoofing può essere evitato attraverso tecniche di autenticazione (Esempio TinySec).
  - SPINS riesce a gestire gli attacchi di replay.

Livello di rete - Selective forwarding or black holes

- I sensori instradano le informazioni verso la base station passando da i nodi intermedi.
- Alcuni nodi intermedi possono essere stati compromessi, i quali possono:
  - inoltrare i pacchetti di dati in modo selettivo;
  - non inoltrare nessun pacchetto ricevuto. In questo modo si crea un black hole.
- **Soluzioni**:
  - Usare multipath routing.

Livello di rete - Sink holes 1

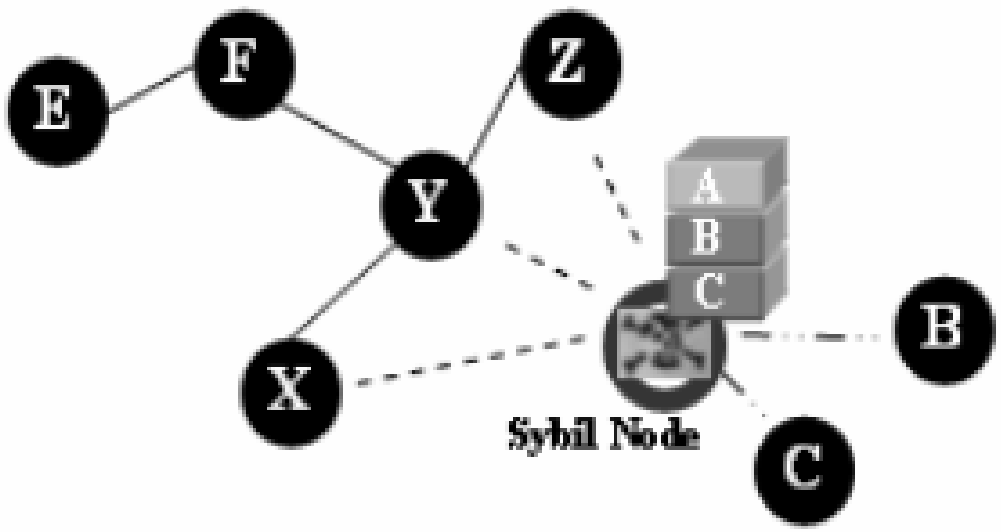
- L'attaccante fa sì che la maggior parte del traffico passi attraverso il nodo compromesso.
- Il nodo malevolo viene reso attraente per l'algoritmo di routing.
  - Un avversario può, ad esempio, falsificare o replicare un advertisement per una route ad alta qualità verso la base station.
- Questo causa un sinkhole con il nodo compromesso al centro.

Livello di rete - Sink holes 2

- Adesso la maggior parte dei dati passano attraverso il nodo compromesso.
- Altri attacchi come wormhole o selective forwarding o eavesdropping possono iniziare attraverso questo attacco.
- **Soluzioni**:
  - Geographic routing: costruisce una topologia su domanda usando solo iterazioni ed informazioni localizzate e senza iniziare dalla base station.
  - Systematic rerouting.

Livello di rete - Sybil attack 1

- Si ha soprattutto nelle reti di sensori che usano la distribuzione di sottoprocessi e la ridondanza di informazioni.
- In questo attacco un nodo malizioso può assumere più identità.



Livello di rete - Sybil attack 2

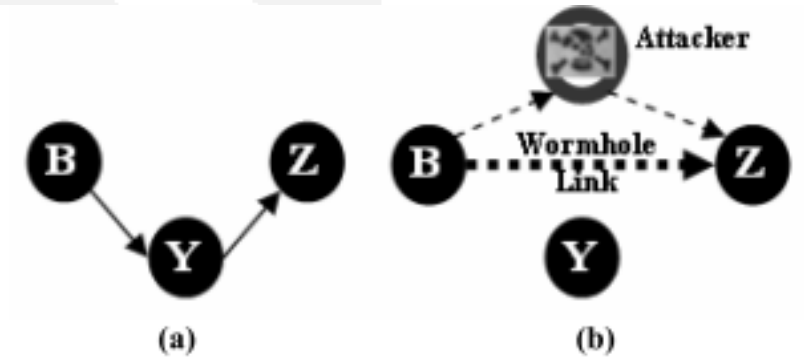
- Può essere eseguito per attaccare gli obiettivi che gli algoritmi distribuiti tentano di raggiungere, come:
  - la memoria distribuita;
  - i meccanismi di routing;
  - l'aggregazione dei dati;
  - il voting;
  - la ripartizione equa delle risorse;
  - la rilevazione dei comportamenti cattivi.
- Il multipath routing non è di aiuto.

Livello di rete - Sybil attack - Cont.

- Può ridurre l'efficacia di meccanismi di tolleranza agli errori.
- **Soluzioni**:
  - Valutare le identità dei nodi:
    - validazione diretta: un nodo fidato valida direttamente l'identità del nodo che si unisce alla rete.
    - validazione indiretta: ad altri nodi fidati è permesso controllare la validità del nodo che si unisce.
  - Verifica della posizione.

Livello di rete - Wormhole attack 1

- Due nodi malevoli creano un canale nascosto tra di loro con cui possono comunicare.
- Un avversario posto vicino ad una base station può distruggere completamente il processo di routing.
  - Se i due nodi sono in differenti posizioni della rete, possono catturano i pacchetti da una parte e ripeterli nell'altra parte della rete.
  - Gli attaccanti possono creare un sinkhole per attrarre i loro nodi vicini.



Livello di rete - Wormhole attack 2

- **Soluzioni**:
- Geographic forwarding: il next hop a cui trasmettere è il nodo più vicino al nodo destinazione. Non esclude del tutto il wormhole.
- Geographic leashes:
  - tutti i nodi sensori devono essere sincronizzati e conoscere la loro posizione corrente.
  - La trasmissione implica un overhead extra per tempo e posizione nel pacchetto da ciascun nodo sorgente nonché le firme digitali.
  - Il ricevente può effettuare delle computazioni predefinite per verificare e autenticare la correttezza del pacchetto.
- Temporal leashes: richiede che solo il timestamp venga trasmesso nel pacchetto.

Livello di rete - Flooding

- Il nodo malevolo può causare parecchio traffico di messaggi inutili nella rete.
- I nodi malevoli possono ripetere alucni messaggi broadcast.
- Causa congestione e può portare ad esaurire tutte le risorse del nodo.
- Attacco di tipo DoS.
- **Soluzioni**:
  - Autenticazione broadcast. Si può usare SPINS, che al suo interno contiene $\mu TESLA$

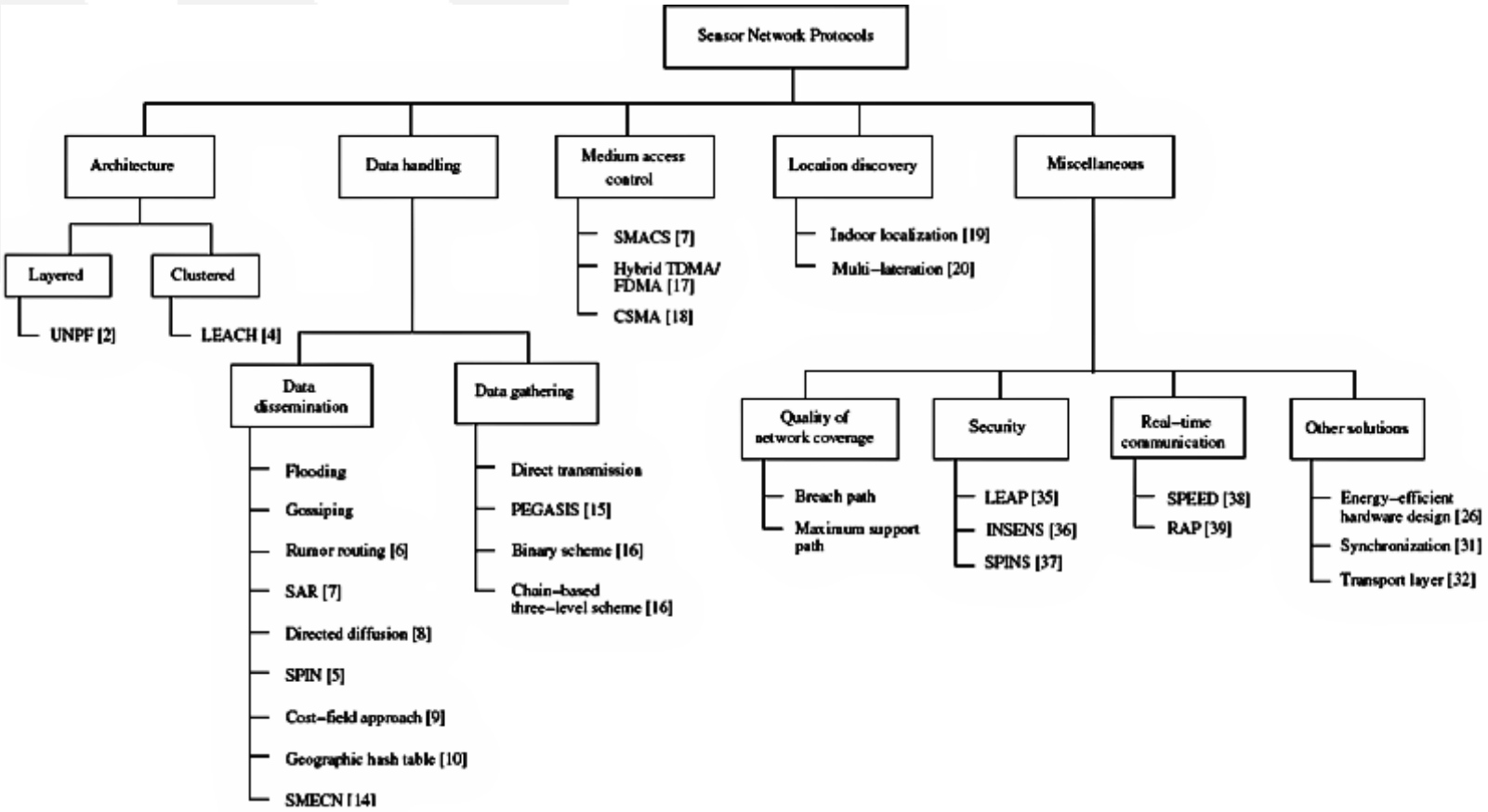
Livello di rete - Attack against privacy

- Eavesdropping e analisi del traffico sono minacce serie per la privacy dei dati delle reti di sensori.
- **Soluzioni**:
  - Meccanismi di anonimato.
  - Privacy policies: definisce il controllo degli accessi e l'autenticazione di ogni altra entità per lo scambio dei dati.

Livello di rete - Hello Flood Attack

- Un attaccante con trasmissione radio elevata invia pacchetti HELLO a un certo numero di nodi sensori che sono distribuiti in una grande aria all'interno di una WSN.
- I sensori pensano che l'avversario sia un loro vicino.
- Il nodo vittima prova quindi a passare per l'attaccante, che può quindi spoofed il nodo.
- **Soluzioni:**
  - Verificare la bidirizionalità di un collegamento.
  - Per ogni nodo autenticare ognuno dei suoi vicini con un protocollo di verifica delle identità usando una base station fidata.

Protocolli WSN in letteratura



Protocolli che vedremo

- **LEAP**
- **SPINS**
- **INSENS**
- **TinySec**
- **Zigbee**

LEAP

LEAP

Localized Encryption and Authentication Protocol - Key management protocol per reti sensori. Supporta in-network processing restrigendo l'impatto di un nodo compromesso solo ai suoi immediati nodi vicini.

- Fornisce meccansimi di keying multipli per fornire confidenzialità e autenticazione.
- Supporta vari pattern di comunicazione (unicast, local broadcast, global broadcast).
- Supporta un protocollo per local broadcast authentication.
  - Si basa su one-way key chains
  - Supporta source authentication senza precludere in-networking processing.

LEAP - Cont.

- È Energy-efficient.
- Evita la frammentazione dei messaggi.
- Minimizza il coinvolgimento della base station.
- Efficente in computazione, comunicazione e memorizzazione.
- Adatto alla difesa dei seguenti attacchi:
  - HELLO Flood attack;
  - Sybil attack;
  - Wormole attack;

LEAP - In-networking processing

In-network processing

Richiede che nodi intermedi possano accedere, modificare e reprimere il contenuto dei messaggi. Esempi di in-network processing sono data aggregation e participazione passiva. Può ridurre il consumo di energia.

- **Participazione passiva**: i sensori prendono determinate azioni in base al messaggio ascoltato.
- Alcuni meccanismi di keying possono precludere o ridurre l'efficacia di in-network processing.
- Per supportare participazione passiva i nodi devono:
  - essere in grado di decifrare o verificare il messaggio scambiato tra due altri sensori.

LEAP - Assunzioni sulla sicurezza

1. Rete di sensori statica
2. La base station agisce da controllore ed è un laptop.
3. I nodi sensori hanno tutte le stesse capacità di comunicazione e computazione.
4. Ogni nodo può memorizzare fino a 100 bytes di keying materials.
5. I nodi non conoscono i loro vicini a priori.
6. L'attaccante può eavesdropping tutto il traffico, iniettare pacchetti o ripetere vecchi messaggi.
7. Se l'attaccante compromette un nodo viene a conoscenza di tutte le informazioni che il nodo possiede.
8. La base station non può essere compromessa.
9. Il livello fisico usa tecniche per difendersi da attacchi di signal jamming.

LEAP - Overwiew

- I pacchetti scambiati dai nodi possono essere classificati in diverse categorie:
  - pacchetti di controllo vs pacchetti di dati;
  - pacchetti broadcast vs pacchetti unicast
  - queries o comandi vs letture del sensore;
- Non tutti i tipi di messaggi hanno gli stessi requisiti di sicurezza:
  - l'autenticazione è richiesta per tutti i tipi di pacchetti.
  - La confidenzialità è richiesta solo per determinati tipi di pacchetti, come le letture di un nodo sensore, le queries inviate dalla base station.
- Meccanismi di single keying non sono appropriati per tutte le comunicazioni sicure richieste.

LEAP - Tipi di chiavi

- LEAP supporta 4 tipi di chiavi:
  - **individual key** che ogni nodo condivide con la base station.
  - **pairwise key** che ogni nodo condivide con un altro nodo sensore.
  - **cluster key** che ogni nodo condivide con più nodi vicini.
  - **group key** che ogni nodo condivide con tutti gli altri nodi nella rete.

LEAP - Individual key

- Ogni nodo ha un'unica chiave condivisa con la base station.
- Esempi di usi:
  - I nodi possono usarla per
    - computare il message authentication code delle letture rilevate se queste devono essere verificate dalla base station.
    - avvisare la base station di comportamenti anomali da parte di nodi vicini.
  - La base station può usarla per cifrare informazioni sensibili da inviare ad un nodo.
- Questa chiave è generata e pre-caricata in ogni nodo prima della suo dispiegamento.
- L'individual key K_u^m per un nodo u è generata da

$$K_u^m = f_{K^m}(u),$$

dove f è un funzione pseudo-calsuale e K^m è una master key.

LEAP - Pairwise key

- Ogni nodo condivide una chiave con ognugno dei suoi vicini immediati.
- Usata per comunicazioni sicure che richiedono privacy o source authentication.
- Preclude la partecipazione passiva.
- Il nodo deve essere in grado di sostenere possibili attacchi almeno per un certo intervallo di tempo.
  - T_{min} tempo necessario ad un avversario per compromettere il nodo sensore
  - T_{est} tempo necessario ad un nodo per scoprire i suoi vicini immediati.
  - $T_{min} > T_{est}$
- 4 passi attraverso cui un nodo u stabilisce le sue pairwise con i nodi vicini.

LEAP - Pairwise key - Passi

1. Key pre-distribution: il controllore genera una chiave iniziale K_I e carica ogni nodo con questa chiave. Ogni nodo u deriva una master key
$$K_u = f_{K_I}(u).$$

2. Neighbor Discovery:
   - u inizializza un timer che scade dopo T_{min}
   - Invia in broadcast un messaggio di HELLO ai suoi vicini contenente il suo id.
   - Attende che i vicini rispondano con un messaggio di ACK. ACK contiene l'id del nodo vicino v ed è autenticato usando la master key $K_v = f_{K_I}(v)$ di v.
   - u conoscendo K_I può derivare K_v e verificare l'identità di v.

LEAP - Pairwise key - Passi

3. Pairwise Key Establishment:
   - u computa la pairwise key con v, come

     $$K_{uv} = f_{K_v}(u)$$

     e lo stesso fa v
   - In questa fase non vi è nessun scambio di messaggi.
4. Key Erasure: quando il tempo scade u cancella K_I e tutte le master keys K_v dei nodi vicini.

LEAP - Cluster Key

- Chiave convidisa da un nodo con tutti i suoi vicini.
- È usata per garantire messaggi broacast a livello locale
  - informazioni di routing control
  - messaggi sicuri che possono beneficiare della partecipazione passiva.
- Permette l'in-networking process.

LEAP - Cluster Key - Cont.

- La fase di cluster establishment segue la fase di pairwise key establishment.
- u genera una random key K_u^c
- Poi cifra questa chiave con la pairwise key condivisa con ogni vicino immediato con cui vuole stabilire una key cluster.
- Trasmette la chiave cifrata ad ogni vicino v_i.
- v_i decifra la chiave, la memorizza in una tabella e trasmette a u la sua cluster key.
- Se un vicino è revocato, u genera una nuova cluster key e la trasmette ai vicini rimanenti.

LEAP - Group key

- Chiave condivisa globalmente usata dalla base station per cifrare messaggi che sono spediti via broascast all'intero gruppo.
- È importante un meccanismo di rekeying efficace per updating questa chiave dopo che un nodo compromesso è stato revocato.
- Ogni nodo viene pre-caricato con la group key.

LEAP - Group key - Cont. 1

- Il rekeying può essere unicast-based: la base station manda una group key ad ogni nodo in modo individuale.
  - Complessità di comunicazione $O(N)$ chiavi, dove N è la dimensione del gruppo.
  - Problema quando N è grande, dovuto alla dimensione dei pacchetti.
  - Non eficciente.
- LEAP usa un sistema di group rekeying basato su cluster keys
- La group key deve essere modificata molto frequentemente.

LEAP - Group key - Cont. 2

- Group rekeying assume l'uso di un protocollo di routing simile a TinyOs beaconing.
- I nodi sono organizzati in in un breadth first spanning tree.
  - La base station manda k'_g a ognuno dei suoi figli nello spanning tree usando la sua cluster key.
  - Un nodo che riceve k'_g verifica l'autenticità di k'_g e quindi la trasmette ai suoi figli nello spanning tree usando la sua cluster key.
  - L'algoritmo procede ricorsivamente.

LEAP - Procedimento di revoca dei nodi 1

- Il meccanismo impiega $\mu TESLA$ per autenticare i messaggi di revoca.
- u è il nodo che deve essere revocato e k'_g è la nuova group key.
  1. La chiave $\mu TESLA$ che deve essere comunicata è k_i^T.
  2. Il controllore invia:
  $$u, f_{k'_g}(0), MAC(k_i^T, u|f_{k'_g}(0))$$
  3. $f_{k'_g}(0)$ è la chiave di verifica che permette a un nodo di verificare l'autenticità della chiave k'_g ricevuta.

LEAP - Procedimento di revoca dei nodi 2

4. Il server di chiavi distribuisce la MAC key k_i^T dopo un intervallo $\mu TESLA$.
5. Il nodo all'intervallo successivo verifica l'aunteticità del messaggio usando la chiave k_i^T.
6. Se la verifica ha successo, il nodo memorizza la chiave di verifica $f_{k'_g}(0)$ temporaneamente.
7. Un nodo v vicino di u rimuove la sua pairwise key condivisa con u e modifica la sua cluster key.

LEAP - Local Broadcast Authentication

- Ogni messaggio nella rete deve essere prima autenticato.
- Lo schema di autenticazione deve essere leggero.
- Local broadcast authentication è necessaria per
  - autenticare messaggi spediti via broadcast localmente
  - supportare partecipazione passiva.
- $\mu TESLA$ non è adatta per local broadcast authentication: non fornisce autenticazione immediata.
- LEAP usa uno schema basato su one-way key chain.

LEAP - Local Broadcast Authentication - Cont. 1

- One-way key chain:
  - ogni nodo genera una one-way key di una certa lunghezza;
  - trasmette il commitment (la prima chiave) della catena ad ogni nodo cifrato con la loro pairwise chiave condivisa;
  - AUTH key: chiave nella catena.
- Autenticazione di messaggio:
  - Attacca al messaggio la successiva AUTH key della key chain (le chiavi sono comunicate in ordine inverso di generazione).
  - Un vicino che riceve il messaggio può verificare il messaggio sulla base del commitment o della AUTH key più recente ricevuta dal nodo mittente.

LEAP - Local Broadcast Authentication - Cont. 2

- Possibile attacco:
  - impedire che il nodo v riceva il pacchetto da u direttamente.
  - L'attaccante può inviare il pacchetto modificato al nodo v impersonando il nodo u.
  - v non ha ancora ricevuto un pacchetto con lo stesso AUTH key quindi accetterà il pacchetto.
- Soluzione: combinare AUTH keys con cluster keys.
  - u calcola una nuova AUTH key mettendo in XOR l'AUTH key con la sua cluster key.

SPINS

SPINS

Security Procolos for Sensor Network - Suite di sicurezza building blocks ottimizzata per ambienti con risorse limitate e dotati di comunicazione wireless. È composto da due building blocks sicuri: SNEP e μ*TESLA*.

- **SNEP**: *Secure Network Encryption Protocol* -protocollo che fornisce confidenzialità dei dati, two-party data authetincation e freschezza dei dati
- μ**TESLA**: *micro Timed, Efficient, Streaming, Loss-tolerant, Authentication Protocol* - protocollo che fornisce autenticazione broadcast per ambienti con risorse limitate.

SPINS - Assunzioni

1. I sensori non sono attendibili.
2. A causa della comunicazione wireless ogni avversario può:
   - eavesdropping il traffico, iniettare nuovi messaggi,ripetere e cambiare vecchi messagi.
3. I messaggi devono essere consegnati alla destinazione con probabilità diversa da 0.
4. Tutti i sensori si fidano della base station:
   - A tempo di creazione, ad ogni nodo viene fornita una master key condivisa con la base station.
   - Tutte le chiavi sono derivate da questa master key.
5. Ogni nodo si fida di sé stesso: in particolare il clock locale deve essere accurato.

SPINS - Progettazione

- A causa dei limiti di computazione SPINS è costruito usando primitive di crittografia simmetrica.
- A causa della memoria tutte le primitive crittografiche sono costruite da unico block cipher.
- Al fine di ridurre l'overhead di comunicazione vengono sfruttate parti comuni tra gli agenti comunicanti.

SPINS - Proprietà di sicurezza

- **Confidenzialità dei dati**:
  - SPINS setta un canale sicuro tra i nodi e la stazione base.
  - Un canale sicuro offre confidenzialità, autenticazione dei dati, integrità e freshness.
- **Autenticazione dei dati**:
  - In two-party communication l'autenticazione si raggiunge puramente con meccanismi simmetrici:
    - mittente e ricevente condividono una chiave;
    - attraverso la chiave viene calcolato un message authentication code (MAC) di tutti i dati comunicati;
    - quando un messaggio con corretto MAC arriva, il mittente sa che deve essere stato spedito dal mittente.
  - Nell'autenticazione broadcast usare un MAC simmetrico non è sicuro.
    - L'autenticazione broadcast deve usare meccanismi simmetrici.
    - $\mu TESLA$ si occupa di questo problema.

SPINS - Proprietà di sicurezza - Cont.

- **Freschezza dei dati**: ogni messaggio deve essere recente.
  - nessun avversario ha replicato vecchi messagi.
  - 2 tipi di freschezza dei dati:
    - Weak freshness: fornisce un ordinamento parziale dei messaggi, ma non fornisce informazioni sul ritardo.
    - Strong freshness: fornisce un ordinamento totale sui messaggi, permette la stima del ritardo.
- **Integrità dei dati**: SPINS raggiunge data integrity attraverso l'autenticazione dei dati.

SPINS - SNEP

- Il protocollo crittografico usa un contatore, il cui valore non viene spedito ma mantenuto in entrambi gli end points.
- SNEP raggiune la semantic security: impedisce che un attaccante riesca a capire il contenuto del messaggio dal ciphertext.
- **Confidenzialità dei dati:**
  - La semplice cifratura non basta, bisogna fornire anche semantic security.
  - SNEP fornisce semantic security senza aggiungere overhead in comunicazione.
    - Si basa su un contatore tra mittente e ricevente per block cipher in counter mode (CTR).
    - Il contatore non necessità di essere inviato con il messagio: i nodi condividono il contatore e lo incrementano ad ogni blocco.

SPINS - SNEP - Cont. 1

- **Two-party authentication e integrità dei dati** si ottiene usando message authentication code (MAC).
- Si usano due chiavi K_{encr} e K_{mac} che sono derivate dalla chiave master secret K.

SPINS - SNEP - Cont. 2

- Fornisce **autenticazione dei dati**: Se il MAC è corretto, un ricevente ha la conferma che il messaggio è stato originato dal mittente dichiarato.
- Fornisce **replay protection**: il valore del contatore nel MAC previene la replica di vecchi messaggi.
- Fornisce **weak message freshness**: se il messaggio è verificato correttamente, il ricevente sa che deve essere stato inviato dopo il messaggio corretto che aveva ricevuto precedentemente.
  - Aggiungendo opportunamente dei nonce (numero casuale lungo abbastanza da essere non predicibile) si può fornire strong freshness.
- Fornisce **basso overhead di comunicazione**: solo 8 bytes addizionali per messaggio, il contatore non viene inviato con il messaggio.

SPINS - $\mu TESLA$

- $\mu TESLA$ cerca di risolvere le ineguatezze di TESLA nei confronti delle reti sensori
  - overhead di comunicazione;
  - uso di firma digitale;
  - one-way-key chain troppo grande.
  - $\mu TESLA$ usa solo meccanismi simmetrici.
  - $\mu TESLA$ rivela la chiave una per intervallo.
  - Poiché è costoso memorizzare una one-way key chain in un nodo sensore $\mu TESLA$ restringe il numero di mittenti autenticati.

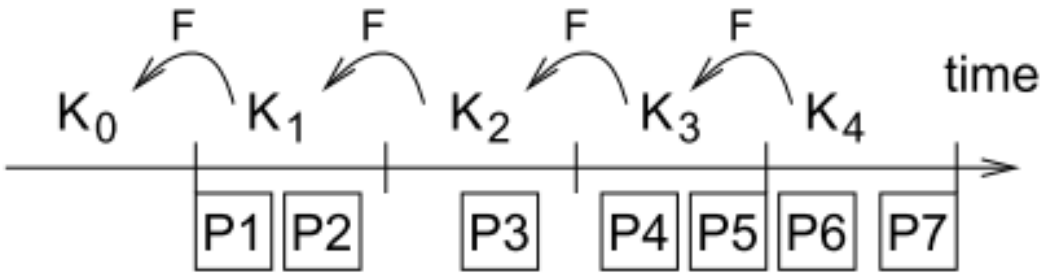
SPINS - $\mu TESLA$ - Overview

- Caso in cui la base station comunica via broadcast informazioni autenticate ai nodi.
  - La base station e i nodi devono essere debolmente sincronizzati.
  - Ogni nodo è a conoscenza di un upper bound sull'errore massimo di sincronizzazione.
  - La base station computa un MAC sul pacchetto con una chiave segreta in quel momento.
  - Quando un nodo riceve un pacchetto, verifica se la corrispondente MAC key non è stata ancora rivelata dalla base station. Sulla base di:
    - debole clock synchronization;
    - massimo errore di sincronizzazione;
    - tempo programmato per rivelare la chiave.

SPINS - $\mu TESLA$ - Overview - Cont.

- Se non già rivelata il nodo memorizza il pacchetto in un buffer
- Quando la chiave deve essere comunicata, la base station manda via broadcast la chiave di verifica a tutti i riceventi.
- Quando un nodo riceve la chiave, verifica la correttezza della chiave.
- Se la chiave è corretta, il nodo può usare il pacchetto autenticato memorizzato nel buffer.
- Ogni chiave MAC è una key chain generata da una pubblica funzione one-way F.
- Per generare la one-way key chain:
  - il mittente sceglie l'ultima chiave K_n della catena in modo random;
  - applica F ripetutamente per calcolare tutte le altre chiavi: $K_i = F(K_{i+1})$
  - Le chiavi sono rivelate in modo indipendente dai pacchetti dei dati.

SPINS - $\mu TESLA$ - Overview - Esempio



- Ogni chiave della key chain corrisponde all'intervallo di tempo.
- Tutti i pacchetti inviati in un intervallo di tempo sono autenticati con la stessa chiave.
- Il ricevente conosce K_0 (commitment della key chain).
- I pacchetti P_1 e P_2 sono inviati nell'intevallo 1 contengono un MAC con la chiave K_1.
- Nessuna chiave è già stata rivelata.

SPINS - $\mu TESLA$ - Overview - Esempio cont.

- Pacchetto P_3 ha un MAC con la chiave K_2 e così via.
- I pacchetti P_4, P_5, P_6 sono persi così come il messaggio che rivela K_1.
- Il ricevente non può quindi ancora autenticare P_1, P_2, P_3.
- Nell'intervallo 4 la base station rivela K_2.
- Il nodo verifica la chiave confrontandola con $K_0 = F(F(K_2))$ e quindi conosce anche $K_1 = F(K_2)$.
- Può ora autenticare P_1, P_2 con K_1 e P_3 con K_2.

SPINS - Implementazione

- L'implementazione è una sfida.
- **Obiettivo**: implementare strong cryptography tenendo conto delle limitazioni hardware.
- **Block cipher**: sottoinsieme di RC5 da OpenSSL.
- **Funzioni di cifratura**: CTR mode. Le stesse funzioni vengono usate sia per codificare che per decodicare.
- **Weak fresheness**: fornita dalla cifratura CTR.
- **Strong freshness**: fornita usando un nonce casuale di 64 bit.
- **Generatore di numeri pseudo-casuali**: funzione MAC con chiave K_{rand} come seed per il generatore.
- **Message authetincation code**: riuso del block cipher, CBC-MAC.

INSENS

INSENS

Intrusion-Tolerant Routing Sensor protocol costruisce tabelle di forwarding in ogni nodo per facilitare la comunicazione tra i nodi sensori e una base station.

- Diminiusce i requisiti di comunicazione, computazione, memoria e larghezza di banda nei nodi sensori.
- Aumenta i requisiti di comunicazione, computazione, memoria e larghezza di banda nella base station.
- INSENS non cerca di rilevare le intrusioni ma di tollerarle, bypassando i nodi malevoli.

INSENS - Cont.

- **Proprietà di INSENS**: anche se un nodo compromette un piccolo numero di nodi a lui vicino, non può danneggiare l'intera rete.
- INSENS deve fornire sicurezza contro:
  - Attacchi DoS che innondano l'intera rete di pacchetti.
  - Attacchi di routing che propagano erronei pacchetti di controllo contenenti false informazioni di routing nella rete.
- Prende vantaggio da una comune architettura delle WSN, l'archiettura assimetrica.
- La base station:
  - funziona come un gateway;
  - ha più risorse in termini di potenza di calcolo, memoria e larghezza di banda rispetto ai nodi.

INSENS - Caratteristiche

1. Per prevenire che la rete venga innondata di pacchetti, il tipo di comunicazione è vincolato.
   - I nodi non possono mandare pacchetti via broadcast all'intera rete.
   - Solo la base station può inviare pacchetti via broadcast.
   - Per i pacchetti unicast, i nodi devono prima passare dalla base station, che agisce da filtro.
2. Per prevenire il routing di dati falsi, le informazioni di control routing devono essere autenticate.
   - La base station ha sempre una parziale conoscenza della topologia della rete.

INSENS - Caratteristiche - Cont. 1

3. La crittografia simmetrica è scelta per confidenzialità e autenticazione tra base station e i nodi;
4. La base station è scelta come punto centrale per la computazione e per la diffusione delle tabelle di routing.
5. Per bypassare i nodi compromessi INSENS usa multipath routing.
   - I percorsi devono essere disgiunti.
   - Se un attaccante compromette un nodo o un percorso, esiste un percorso secondario per inoltrare i pacchetti.

INSENS - Caratteristiche - Cont. 2

6. Ogni nodo condivide una chiave segreta solo con la base station, non con gli altri nodi.
   - Se un nodo è compromesso, l'attaccante può accedere solo a una chiave segreta.
   - Non accede alle chiavi dei vicini e/o agli altri nodi nella rete.
7. Ogni nodo necessità di essere programmato con una sola chiave segreta per farlo autenticare con la base station, e una chiave iniziale per autenticare la base station ad ogni nodo.

INSENS - Principi di design

1. Sfruttare la ridondanza per tollerare le intrusioni, non richiede che l'intrusione venga rilevata;
   - Rilevare le intrusioni è costoso e difficile.
   - INSENS è robusto in presenza di un piccolo numero di avversari.
2. Eseguire tutte le computazioni pesanti nella base station, minimizzando il ruolo dei sensori.
3. Limitare la portata di un danno provocato da un avversario limitando il flooding e usando meccanismi di autenticazione.
   - Usa un meccansimo di autenticazione one-way per autenticare le informazioni inviate dalla base station.
   - Usa meccansimi di integrità per assicurare che eventuali manomissioni vengano rilevate dal destinatario.
   - I nodi possono scartare i messaggi duplicati.

INSENS - Principi di design

- **Ridondanza nel routing**:
  - Permette di bypassare gli intrusi quando instradano i messaggi.
  - Ogni percorso è indipendete l'uno dall'altro, condividono il minor numero possibile di nodi/collegamenti.
  - Idealmente solo sorgente e destinazione dovrebbere essere comuni a più path.
  - Ogni messaggio è inviato più volte, una volta per ciascun percorso rindondante.
  - Se almeno un percorso non è compromesso, una copia del messaggio arriverà a destinazione.
  - Opportuni meccanismi di confidenzialiàtà, integrità e autenticazione sono necesssari affinché un destinatario possa determinare quale copia ricevuta è originale e quale è contraffata.

INSENS - Route Discovery

- Permette di determinare la topologia della rete di sensori
- Costruisce l'appropriata tabella di forwarding per ogni nodo.
- È eseguita in 3 round:
  - **Route Request**
  - **Route Feedback**
  - **Routing Table Propagation**

INSENS - Forwarding Data

- Usando la forwarding tables costruita nella fase di route discovery: i dati sono inoltrati dal nodo sorgente alla base station e dalla base station al nodo destinazione.
- Un nodo mantiene una forwarding table con parecchie voci, una per ogni strada a cui il nodo appartiene.
- Ogni entry è 3-upla: destinazione, sorgente, mittente immediato.
  - Mittente immediato è l'id del nodo che ha appena inoltrato il pacchetto.
- Quando un nodo riceve un pacchetto cerca una voce corrispondente nella sua forwarding table. Se la trova trasmette il pacchetto.

TinySec

TinySec

TinySec è un'architettura di sicurezza link layer interamente implementata per reti di sensori wireless. È un generico leggero pacchetto di sicurezza facilmente integrabile.

- I software basati su protocolli link layer sono:
  - efficenti;
  - aggiungono meno del 10% di energia, latenza e overhead nella dimensione di banda.
- TinySec si basa su primitive che altri ricercatori hanno dimostrato essere sicure.
- Con queste primitive viene realizzato un protocollo di sicurezza link layer leggero ed efficente.

TinySec - Motivi per sicurezza link layer

- Nelle reti convenzionali la sicurezza è raggiunta da meccanismi end-to-end.
  - I router intermedi devono solo vedere gli headers dei pacchetti non preoccupandosi del corpo del messaggio.
- Nelle reti di sensori:
  - In-networking processing è utile per eseguire aggregazione di messaggi ed eliminazione di messaggi duplicati.
- Meccanismi di sicurezza link layer garantiscono autenticazione, integrità e confidenzialità dei messaggi tra nodi vicini, fornendo in-networking processing.

TinySec - Motivi per sicurezza link layer - Cont.

- Meccansimi end-to-end sono vulnerabili a certi tipi di attacchi DoS:
  - Se l'integrità è controllata solo alla destinazione finale, la rete può instradare pacchetti iniettati da un avversario molti hop prima di dove vengono rilevati.
  - Causando perdità di energia e larghezza di banda.
- Meccanismi link layer:
  - Rivelano pacchetti non autorizzati quando sono iniettati.
- Meccanismi end-to-end possono possono essere usati da complemento di TinySec.

TinySec - Obiettivi

1. Sicurezza
2. Performance
3. Usabilità

TinySec - Sicurezza

- TinySec deve garantire:
  - Controllo degli accessi
  - Integrità del messaggio
  - Confidenzialità del messaggio

TinySec - Sicurezza Cont. 1

- TinySec non offre protezione da Replay attack.
  - Una protezione comune è includere un contatore che incrementa ad ogni messaggio.
  - Messaggi con contatori passati vengono rifiutati.
  - Il destinatario deve mantenere una tabella con l'ulitmo contatore ricevuto per ogni mittente.
  - Questa tabella si può riempire facilmente.
  - Se la tabella si riempie il destinatario può
    1. ignorare i messaggi dei mittenti che non sono nella tabella dei vicini.
    2. pulire la tabella.

TinySec - Sicurezza Cont. 2

- Nessuna delle due soluzioni è accettabile: causano rispettivamente DoS attacks e replay attacks.
- Il livello applicazione è più adatto per gestire le tabelle di replay.
- Alcune informazioni come la topologia della rete non sono disponibili nel livello link.

TinySec - Performance

- TinySec mira a:
  - Un consumo moderato di energia.
  - Un uso di canali e di latenza comparabili a quelli di una rete di sensori senza meccansimi di sicurezza.
  - Fornire una protezione ragionvelo limitando l'overhead.

TinySec - Usabilità

- **Security platform**:
  - I protocolli a livello più alto devono essere in grado di usare TinySec come primitiva.
  - TinySec deve fornire il giusto insieme di interfacce affinché i livelli più alti non devano implementare meccanismi già disponibili.
- **Trasparenza**:
  - TinySec deve essere trasparente alle applicazioni eseguite su TinyOS.
  - Deve però garantire allo stesso tempo una modalità di personalizzazione della sicurezza semplice.

TinySec - Usabilità - Cont.

- **Portabilità**
  - TinySec deve essere portabile, si deve adattare a diverse piattaforme e architetture radio.
  - TinyOS viene eseguito su differenti processori:
    - Texas Instruments
    - Atmel
    - Intel x86
    - StrongArm
  - TinyOS usa due architettetture radio:
    - Chipcon CC1000
    - RFM TR1000

TinySec - Design

- TinySec supporta due differenti opzioni di sicurezza:
  1. TinySec-AE che fornisce crittografia autenticata: cifra il payload dei dati e autentica il pacchetto usando MAC.
  2. TinySec-Auth fornisce solo autenticazione usando MAC.
- Il MAC è calcolato sui dati cifrati e sull'header del pacchetto.

TinySec - Design

- IV a 8 byte
- Cipher block chaining Skipjack
- TinySec sempre autentica i messaggi, mentre la confidenzialità è opzionale.
- TinySec usa CBC-MAC per calcolare e verificare i MAC.

Analisi - Integrità e autenticazione di messaggio

- La sicurezza di CBC-MAC è collegata alla lunghezza del MAC.
- TinySec usa un MAC di 4 byte.
- Si può dimostrare che tale lunghezza non è dannosa per le reti di sensori.
  - Un attaccante dovrebbe inviare in media 2^{31} pacchetti prima che possa falsificare con successo il MAC di un singolo pacchetto.
  - Su canali di 19.2 kb/s si possono solo inviare 40 tentativi per secondo.
  - Per inviare 2^{31} pacchetti servirebbero 20 mesi.
  - La batteria di un nodo non può supportare la ricezione di così tanti messaggi.

Analisi - Confidenzialità 1

- TinySec usa un IV a 8 byte.
- I primi 4 byte sono $dst|AM|l$.
- Gli utimi 4 byte sono $src|ctr$.
- *dst* è l'indirizzo di destinazione, *AM* è il tipo di messaggio attivo, *l* è la lunghezza del payload di dati, *src* è l'indirizzo sorgente e *ctr* è il contatore di 16 bit.
- Il formato degli ultimi 4 byte cerca di massimizzare il numero di pacchetti che ogni nodo può inviare prima che ci sia una ripetizione globale di IV.
- Ogni nodo può inviare almeno 2^{16} pacchetti prima che l'IV venga riusato.

Analisi - Confidenzialità 2

- Nel caso di 1 pacchetto al minuto per nodo, il riuso di IV avverrà dopo 45 giorni.
- Il riuso di IV è un problema solo se viene usato con la stessa chiave. Prima di riusarlo è quindi opportuno prevedere un aggiornamento della chiave.
- In caso di IV ripetuto l'informazione è persa quando un nodo invia due pacchetti differenti con gli stessi primi 8 bytes, IV e $dst|AM|I$.

Analisi - Meccanismi di gestione delle chiavi

- Qualunque protocollo di gestione delle chiavi può essere usato con TinySec
- La chiave in TinySec è una coppia di chiavi Skipjack: una per i dati e una per il MAC.

TinySec - Implementazione 1

- TinySec gira su piattaforme Mica (radio RFM TR1000), Mica2 e Mica2Dot (radio Chipcom CC1000) tutti con processore Atmel.
- È stato integrato nel simulatore TOSSIM, che gira su Intel X86.
- Altri hanno portato TinySec su microprocessori Texas Instruments.
- È facilmente portabile sia su nuovi processori che su nuove architettetture radio.
- TinySec è stato implementato su 3000 linee di codice nesC.
- L'implementazione richiede 728 bytes di RAM e 7146 bytes di program space.

TinySec - Implementazione 2

- Lo stack radio di TinyOS 1.1.2 è stato modificato per permettere che le operazioni crittografiche di TinySec vengano eseguite in tempo.
- È stato implementato sia con RC5 che con Skipjack.
- È stato implementato un modello di condivisione di chiave a livello network, con distribuzione a tempo di compilazione.
- Si abilita specificando "TINISEC=true" nella linea di comando del make.
- TinySec è distribuito con la realese officiale di TinyOS.

Valutazioni

- TinySec aumenta i costi relativi a computazioni e consumo energetico nell'invio di un pacchetto.
- I motivi di questo aumento sono:
  - la dimensione più grande del pacchetto.
  - la computazione extra necessaria per la crittografia.
- La lunghezza del pacchetto varia da 1 a 5 bytes (a seconda se si usa TinySec-Auth o TinySec-AE).
- Pacchetti più lunghi:
  1. Riducono la larghezza di banda.
  2. Incrementano la latenza.
  3. Incrementano il consumo di energia perché la radio deve stare accesa di più.

Zigbee

Zigbee

Protocollo sviluppato dalla Zigbee Alliance al fine di raggiungere uno standard di comunicazione wireless a due vie, basso costo e consumo.

- Lo stack architecture è composto da un insieme di layers ognuno dei quali esegue una serie di servizi per il livello sottostante.
- Due entità:
  - **Entità di dati** che fornisce un servizio di trasmissione dei dati;
  - **Entità di gestione** che fornisce tutti gli altri servizi

Zigbee - Cont.

- Ogni entità espone un'interfaccia verso il livello superiore attraverso un *Service Access Point (SAP)*.
- Ogni SAP supporta un determinato numero di primitive al fine di raggiungere le funzionalità richieste.

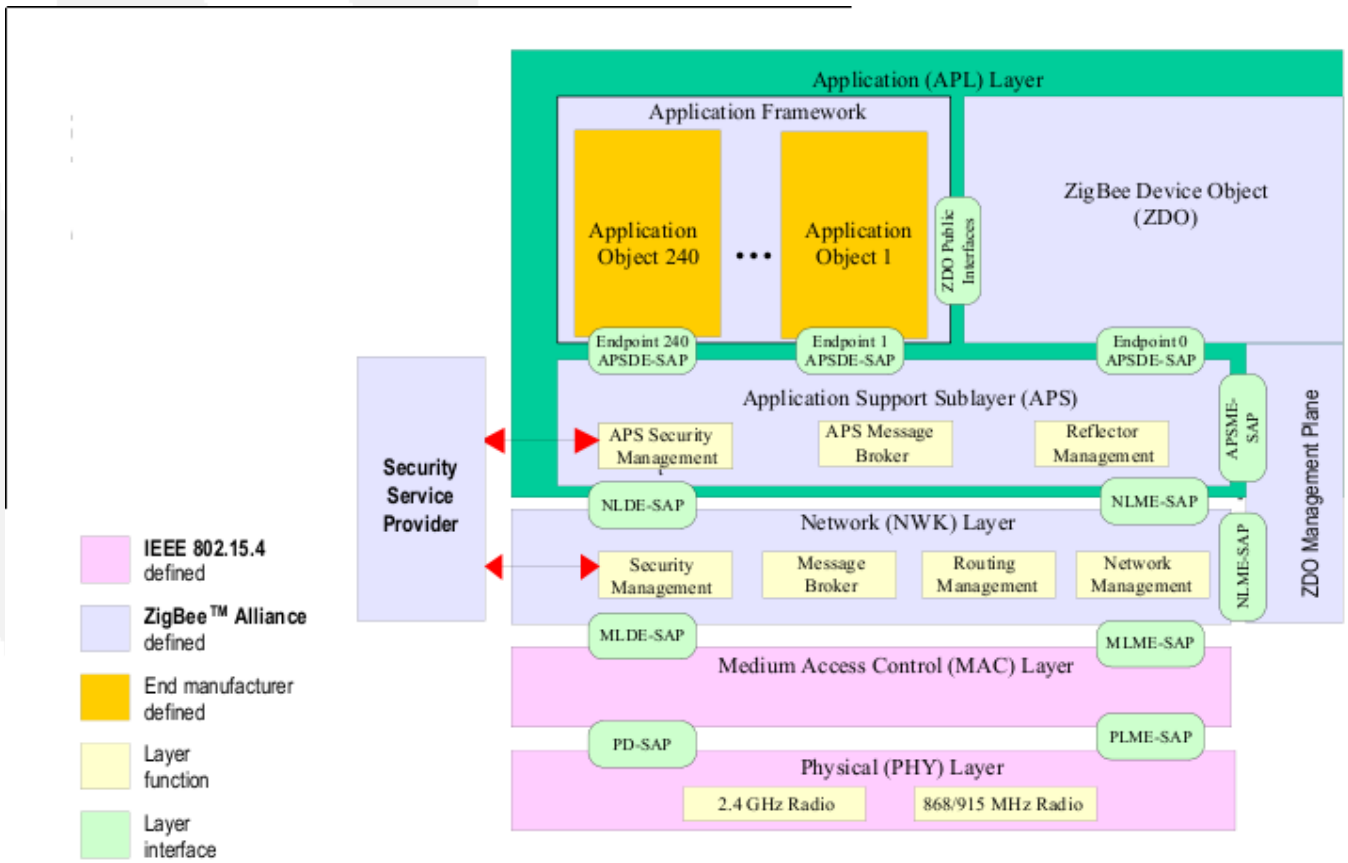
Caratteristiche ZigBee 1

- Assicura un uso efficiente della larghezza di banda da parte dei dispositivi.
- Trasmette messaggi in maniera affidabile.
- Non garantisce la consegna dei messaggi.
- Ci sono tre tipi di dispositivi:
  1. Coordinatore
  2. Router
  3. End Device

Caratteristiche ZigBee 2

- Sono disponibili 3 tipi di topologie:
  1. Star
  2. Tree
  3. Mesh
- Sicurezza
  - A chiavi simmetriche AES-128
  - Autenticazione e Crittografia ai livelli MAC, NWK e Applicazione.
  - Gerarchia di chiavi:
    - Master keys
    - Network keys
    - Link keys

ZigBee - Stack Architecture 1



ZigBee - Stack Architecture 2

- 2 livelli sono definiti dallo standard IEEE 802.15.4-2003:
  - Livello fisico (PHY): IEEE 802.15.4-2003 ha 2 livelli fisici che operano su due frequenze separate:
    - 868/915 MHz
    - 2,4 GHz
  - Sottolivello medium access control (MAC): controlla l'accesso al canale usando un meccanismo CSMA-CA. Si occupa:
    - di trasmettere beacon frame;
    - della sincronizzazione;
    - di fornire un meccanismo di trasmissione affidabile

ZigBee - Stack Architecture 3

- I livelli forniti dalla ZigBee Alliance sono:
  - Livello di rete (NWK)
  - Livello applicazione (APL).
    - È formato dall'*Application Support Sublayer(APS)* dallo *Zigbee device objects (ZDO)* e dall'Application Framework.
    - Gli *Application Objects* definiti dai produttori usano il framework applicazione e condivono l'APS e i servizi di sicurezza con lo ZDO.

ZigBee - Livello Applicazione

- **Application Support Sub-Layer**:
  - fonisce un'interfaccia tra il livello di rete (NWk) e il livello applicazione (APL).
  - Applica o rimuove il livello di sicurezza.
  - Supporta i pacchetti data, comando e ACK.
- **Application Framework**: ambiente in cui le *Application objects* sono ospitate in un dispositivo ZigBee. Possono essere definiti fino a 240 distinte *application objects*.
- **ZigBee Device Object**: fornisce un'interfaccia tra le *application objects*, il device profile e l'APS.
  - Si trova tra l'application framework e l'APS.
  - Inizializza l'APS, il livello di rete (NWK) e il Security Service Provider.
  - Assembla le informazioni di configurzione dell'applicazione per determinare ed implementare discovery, gestione della sicurezza, gestione della rete e binding management.
  - Gestisce in modo intelligente le primitive a livello NWK.

Livello di Rete 1

- Assicura il corretto funzionamento del sotto-livello MAC IEEE 802.15.4-2003.
- Fornisce un'interfaccia al livello applicazione.
- L'entità dati del livello NWK fornisce:
  - Generazione del Network level protocol data unit (PDU).
  - Routing per la topologia specifica.
  - Sicurezza

Livello di Rete 2

- L'entità di gestione fornisce:
  - Configurazione di un nuovo dispositivo
  - Inizializzazione di una nuova rete
  - Joining, Rejoining e leaving di una rete.
  - Indirizzamento: ZC o ZR assegnano gli indirizzi ai dispositivi che si uniscono alla rete.
  - Scoperta dei vicini
  - Route discovery
  - Reception control: abilità di un dispositivo di controllare quando il ricevitore è attivato e per quanto tempo.
  - Routing

ZigBee - Sicurezza

- ZigBee fornisce autenticazione, crittografia, freshness (frame counters) e integrità di messaggio.
- I Servizi di sicurezza offerti sono:
  - Key establishment
  - Key transport
  - Frame protection
  - Device managment
- 2 modi di sicurezza:
  - Residential mode
  - Commercial mode

ZigBee - Sicurezza - Assunzioni 1

- I protocolli di sicurezza vengono eseguiti correttamente nella sua interezza.
- La generazione dei numeri casuali avviene come ci si aspetta.
- Le chiavi segrete non sono disponibili in modo non sicuro fuori dal dispositivo
  - nessun dispositivo trasmette né intenzionalmente né innavvertitamente il suo materiale di codifica ad altri device a meno che il materiale non sia stato protetto.
  - Solo quando un dispositivo che non è ancora stato pre-configurato si unisce alla rete, la chiave può essere inviata senza protezione.

ZigBee - Sicurezza - Assunzioni 2

- **Open trust model**: differenti livelli dello stack e tutte le applicazioni in esecuzione su un singolo device si fidano l'una dell'altra.
  - I servizi di sicurezza proteggono in modo crittografico le interfacce solo tra dispositivi diversi.
  - La separazione di interfacce tra differenti livelli dello stack sullo stesso dispositivo è organizzata in modo non crittografico, attraverso una corretta progettazione della sicurezza dei SAP.
  - Permette di riusare lo stesso materiale di codifica tra differenti livelli dello stesso dispositivo
  - Permette di realizzare la sicurezza end-to-end su base device-device anziché tra coppie di livelli o applicazioni su due dispositivi comunicanti.

ZigBee - Sicurezza - Architettura

- Il livello che origina un frame è responsabile della sua sicurezza inziale.
- L'open trust model diminuisce i costi di memorizzazione.
- La sicurezza è end-to-end: solo i dispositivi sorgente e destinazione accedono alle loro chiavi condivise.
  - Bisogna fidarsi di solo due dispositivi.
  - Il routing dei messaggi è realizzato indipendentemente da considerazioni sulla fiducia.
- Il livello di sicurezza usato da tutti i dispositivi di una rete e da tutti i livelli di un dispositivo deve essere lo stesso.

ZigBee - Sicurezza - Chiavi 1

1. Link keys:
   - garantisce comunicazioni unicast tra entità peer APL
   - È a 128 bit
   - È condivisa solo tra due dispositivi
   - Vine usata solo da APS.
2. Network Keys:
   - garantisce comunicazioni broadcast tra i dispositivi;
   - È a 128 bit
   - È condivisa tra tutti i dispositivi
   - È di due tipi: standard e ad alta sicurezza. Il tipo di chiave determina il modo in cui la chiave è distribuita e come i contatori dei frame vengono inizializzati.
   - Viene usata dai livelli NWK e APL.

ZigBee - Sicurezza - Chiavi 2

- Un dispositivo acquisisce le link keys in 3 modi:
  1. Key transport
  2. Key-estbalishment: si basa su una master-key che viene acquisita o attraverso il key-transport o attraverso la pre-installazione. La master key viene usata solo da APS.
  3. Pre-installazione
- Un dispositivo acquisisce la network keys in 2 modi:
  1. Key transport
  2. Pre-installazione
- Servizi di sicurezza differenti devono usare chiavi differenti.

Zigbee - NWK layer security

- Il meccanismo di frame protection deve essere usato se:
  - il frame è generato a livello NWK
  - il frame è generato a un livello più alto ma l'attributo *nwkSecureAllFrames* è a TRUE
  - il parametro SecurityEnable è diverso da FALSE.
- Questo meccanismo fa uso di AES e di CCM*.
- I livelli superiori gestionco la sicurezza del livello NWK
  - creando le network keys attive.
  - determinando quale livello di sicurezza usare.

ZigBee - APL Layer Security

- Se un frame generato a livello APL deve essere sicuro, la sicurezza viene gestita dal sottolivello APS.
- Il livello APS può usare sia link keys che network keys e fornisce i servizi per mantenere e stabilire le relazioni di sicurezza.
- Lo ZDO gestisce le politiche di sicurezza e la configurazione di sicurezza di un dispositivo.

ZigBee - APL Layer Security - Key establishment 1

- Sono realizzati dal sottolivello APS.
- Fornisce il meccanismo attraverso cui un dispositivo ZigBee condivide una Link Key con un'altro dispositivo.
- Viene preceduto da un passo di trust-provisioning.
- Le trust information (es master key) sono il punto di partenza per stabilire una link key.
- Le trust information possono essere fornite in-band o out-of-band.

ZigBee - APL Layer Security - Key establishment 2

- Protocollo di Key-Establishment:
  1. Scambio di dati effimeri
  2. Uso dei dati effimeri per derivare link key
  3. Conferma che la link key è stata calcolata correttamente.

ZigBee - APL Layer Security - Key establishment 3

- Procollo SKKE (Symmetric-Key Key Establishment)
  - Dispositivo iniziatore stabilisce una link key con dispositivo rispondente usando una master key.
  - La master key può essere:
    - pre-installata durante la fase di produzione
    - installata da un Trust Center
    - si può basare su dati inseriti dall'utente.
  - La master key deve essere mantenuta segreta e autenticata.

ZigBee - APL Layer Security - Transport Key

- Fornisce mezzi sicuri e insicuri per trasportare una chiave da un dispositivo all'altro.
- Il comando sicuro fornisce un mezzo per trasportare master key, link key o network key da una source key agli altri dispositivi.
- Il comando non sicuro fornisce un mezzo per caricare un dispositivo con una chiave iniziale. Non protegge la chiave crittograficamente.

ZigBee - APL Layer Security - Update device

- Fornisce un mezzo sicuro attraverso cui un dispositivo informa un secondo dispositivo che un terzo dispositivo ha fatto un cambio di stato e deve essere aggiornato.
- Il primo dispositivo è per esempio un Trust Center.
- Il secondo dispositivo è per esempio un router.
- Permette al Trust Center di mantenere una lista di dispositivi attivi nella rete.

ZigBee - APL Layer Security - Remove device

- Fornisce un mezzo sicuro attraverso cui un dispositivo informa un altro dispositivo che uno dei suoi figli deve essere rimosso dalla rete.
- Il primo dispositivo è per esempio un Trust Center.
- Il secondo dispositivo è per esempio un router.
- Lo si può usare per rimuovere un dispositivo che non soddisfa i requisiti di sicurezza del Trust Center.

ZigBee - APL Layer Security - Request key

- Mezzo sicuro con cui un dispositivo può richiedere la network key attiva, o una master key end-to-end da un altro dispositivo (esempio Trust Center).

ZigBee - APL Layer Security - Switch key

- Mezzo sicuro con cui un dispositivo (es. Trust Center) può informare un altro dispositivo che dovrebbe passare ad una diversa network key attiva.

ZigBee - APL Layer Security - Entity Authentication

- Mezzo sicuro con cui un dispositivo può sincronizzare le informazioni con un altro dispositivo.
- Fornisce simultaneamente autenticazione in base ad una chiave condivisa.

ZigBee - APL Layer Security - Permissions Configuaration table

- Indica quali dispositivi hanno l'autorizzazione di effettuare certi tipi comandi.
- Determina sempre se è richiesta o meno la link key.

ZigBee - Trust Center role 1

Trust Center

Dispositivo di cui gli altri dispositivi all'interno di una rete si fidano per la distribuzione delle chiavi e per la gestione delle configurazioni di applicazione end-to-end.

- Ci deve essere esattamente un Trust Center per ogni rete.
- Tutti i membri di una rete dovrebbero riconoscere almeno un Trust Center.
- Tipicamente è lo ZC.

ZigBee - Trust Center role 2

- Nelle applicazioni in Commercial mode che richiedono elevata sicurezza:
  - un dispositivo può essere pre-caricato con un indirizzo di Trust center e con una master key.
  - se è tollerato un momento di vulnerabilità, la master key può essere inviata tramite un servizio di key transport insuro in-band.

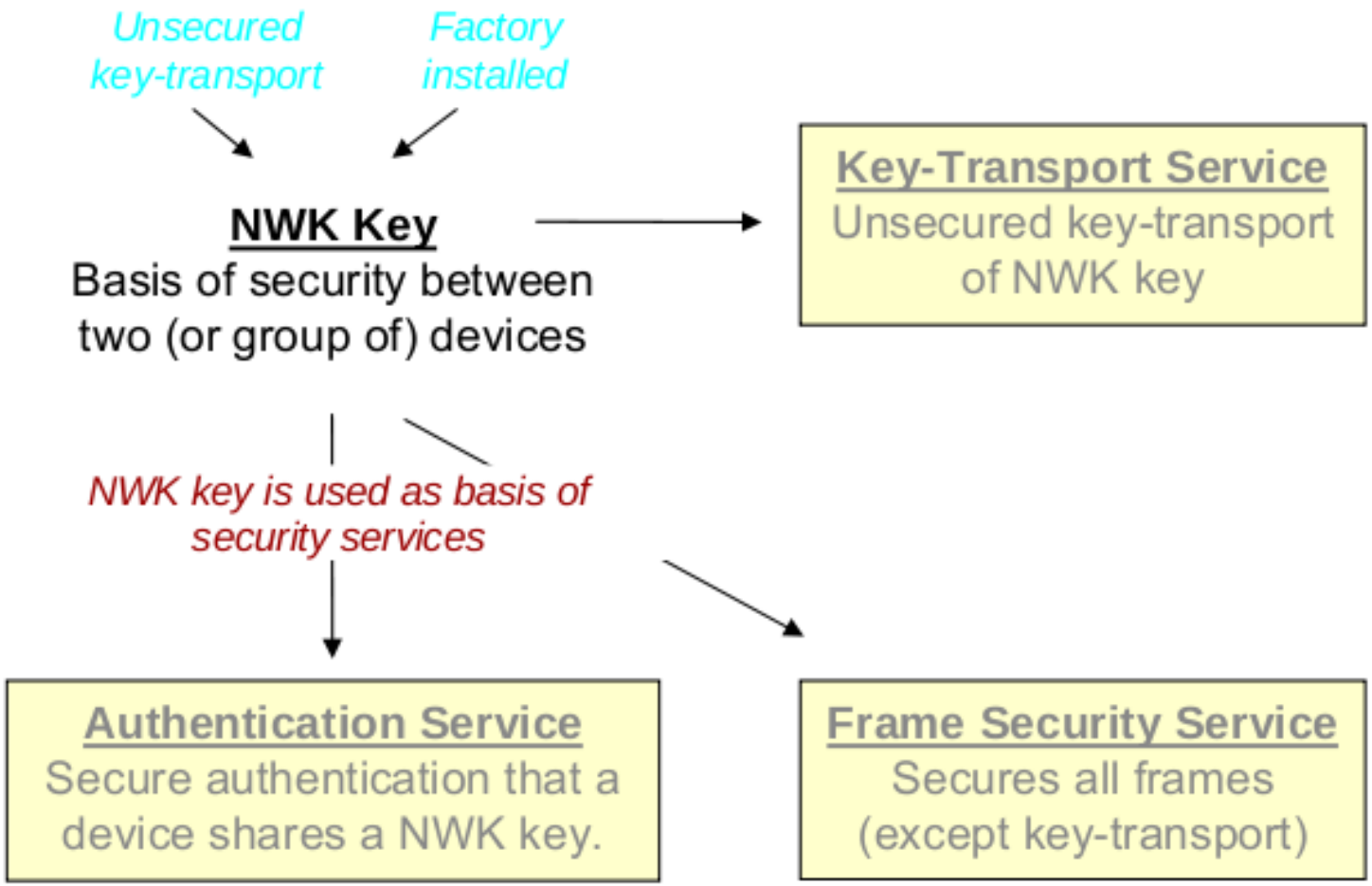
ZigBee - Trust Center role 3

- Nelle applicazioni in Residential mode che richiedono bassa sicurezza:
  - un dispositivo comunica in modo sicuro con il Trust Center usando la network key corrente.
  - La network key può essere pre-configurata o inviata attraverso un servizio di key transport insicuro in-band.

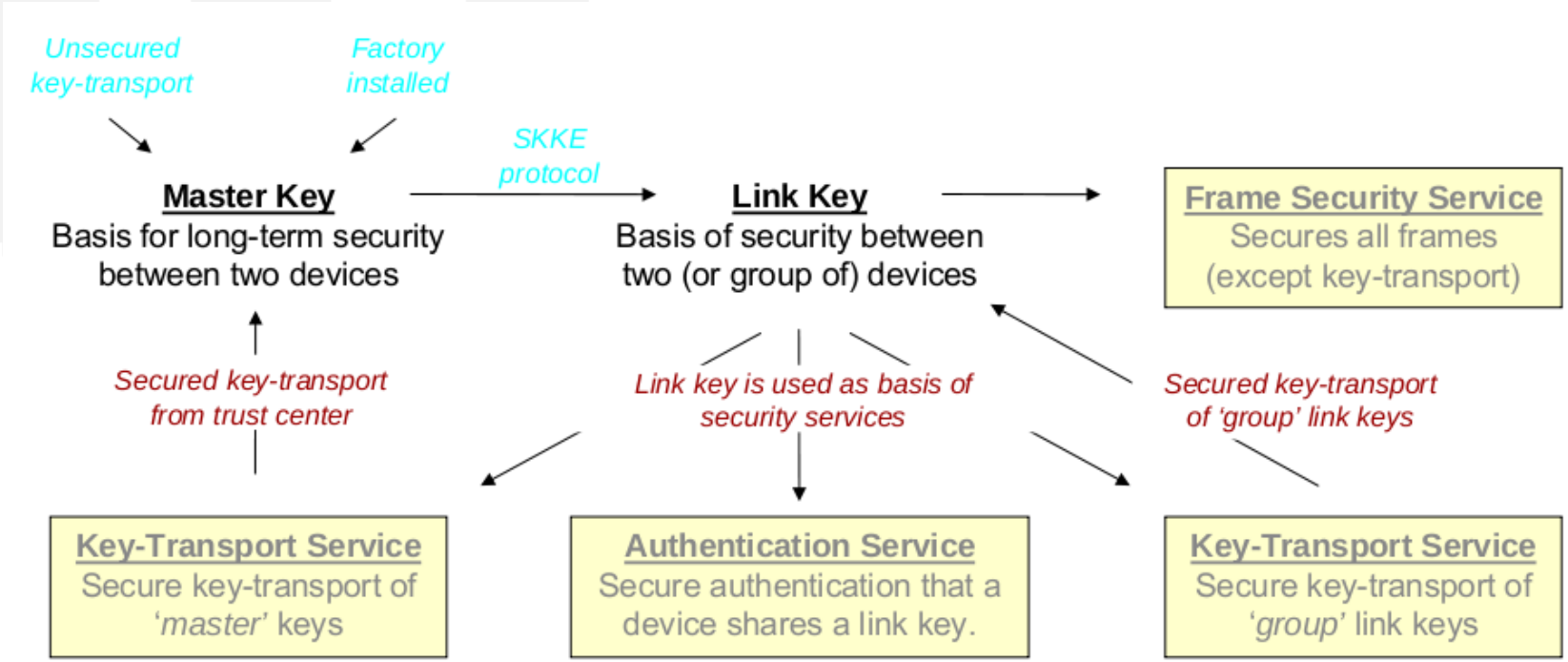
ZigBee - Trust Center role 4

- Per scopi di trust management: un dispositivo accetta una master key o una network key attiva originata dal Trust Center attraverso un servizio di key transport insicuro.
- Per scopi di network managment: un dispositivo accetta una key network attiva iniziale e aggiorna le network keys solo dal suo Trust Center.
- Per scopi di configurazione: un dispositivo accetta un master key o una link key solo dal suo Trust Center.
- Le chiavi non iniziali di link, master o network sono accettate solo se originate dal Trust Center attraverso un servizio di key transport sicuro.

Servizi di sicurezza in Residendial Mode



Servizi di sicurezza in Commercial Mode



Riferimenti bibliografici

- I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci *A Survey on Sensor Networks* IEEE Communications Magazine 2002
- Z.S. Bojkovic, B.M. Bakmaz, M.R. Bakmaz *Security Issues in Wireless Sensor Networks* International Journal of Communications, Issue 1, Volume 2, 2008
- J. Deng, R. Han, S. Mishra *INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks* Science
- A. Habib *Sensor Network Security Issues at Network Layer* Networks, Islamabad 2008

Riferimenti bibliografici

- F. Hu, J. Ziobro, J. Tillett, N.K. Sharma *Secure Wireless Sensor Networks: Problems and Solutions* Cybernetics
- C. Karlof, N. Sastry, D. Wagner *TinySec: A Link Layer Security Architecture for wireless Sensor Networks* Networks
- C. Karlof, D. Wagner *Secure routing in wireless sensor networks: attacks and countermeasures* Ad Hoc Networks, 2003
- A. Pathan *Security in wireless sensor networks: issues and challenges* 2006 8th International Conference Advanced Communication Technology, 2006

Riferimenti bibliografici

- A. Perrig, R. Szewczyk, V. Wen, D. Culler, J.D. Tygar *SPINS: Security Protocols for Sensor Networks* Electrical Engineering, 2001
- S. Zhu, S. Setia, S. Jajodia *LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks* Information Systems
- ZigBee Alliance *ZigBee Specification* Gennaio 17, 2008
- ZigBee Alliance *ZigBee Architecture Overview* Milan March 2006

Riferimenti bibliografici


A.S. Tanenbaum *Computer Networks* International Edition, Fourth Edition


L. Viganò *Introduction* Dipartimento di Informatica, Università di Verona 2011