Studienstiftung
des deutschen Volkes

Unidirectional Key Distribution Across Time and Space with Applications to RFID Security [JPP08]

Studienstiftung des deutschen Volkes
Summer Academy La Colle-sur-Loup 2009

Sandro Bauer

Studienstiftung des deutschen Volkes

Outline (1)

- Introduction
  - Key exchange as a basic problem of cryptography
  - Special characteristics of RFID supply chains
  - Gen2 features requiring secret key exchange
  - Object hierarchies in RFID supply chains

Studienstiftung des deutschen Volkes

Outline (2)

- Secret sharing across space
  - General idea and terminology used
  - Main advantages compared to other key distribution schemes used in RFID environments
  - Space constraints on tags and their implications
  - TSS schemes compared to related work
  - Definition of TSS schemes
  - Definition of privacy and robustness experiments
  - Illustration and formal definitions
  - Juels' implementation sketch and real world parameterization

Studienstiftung des deutschen Volkes

Outline (3)

- Secret sharing across time
  - Why use time-based secret sharing?
  - General explanation and some examples
  - The SWISS model: Basics and terminology
  - Definition of (k,n)-SWISS schemes
  - Making share sizes independent of secret size
  - Brief presentation of Generic SWISS Families

- Outlook on future work

Introduction (1)

- Key management and revocation is one of the basic problems of cryptography
  - "Alice and Bob"
  - Restrictions inherent to manual key distribution systems
  - Secure channels as the sole means for ensuring privacy
- Special characteristics of RFID supply chains
  - Unidirectional channels
  - Possible anonymity of widespread and multi-hierarchical supply chains
- Gen2 features requiring secret key exchange
  - Locking, perma-locking and kill commands

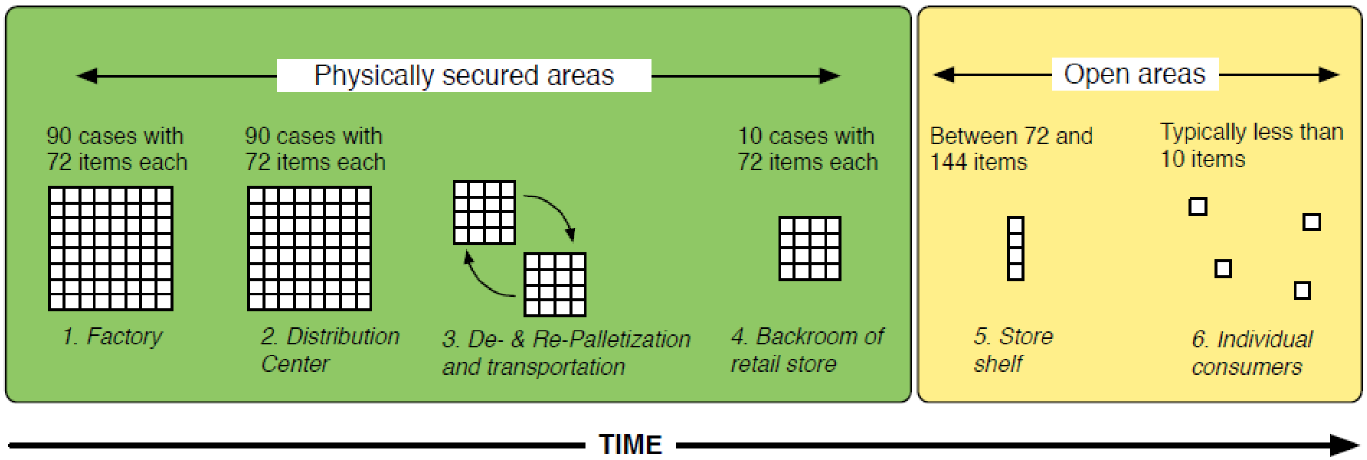
Introduction (2)

- Motivation: Highly critical security issues with RFID-labeled products at item level
  - Ultimate sellers are burdened with the duty of invoking killing commands on each sold product
    - Retailers are likely to lack required technical infrastructure
    - Problems with ensuring provability of killing actions
  - Reliable and secure distribution of kill PINs to the retailer required
  - Serious security risks due to possibly unreliable (or criminal) retailers
    - Problems with ensuring tag integrity
    - Possible misuse of RFID tags by retailers

Studienstiftung des deutschen Volkes

Introduction (3)

- Object hierarchies in RFID supply chains



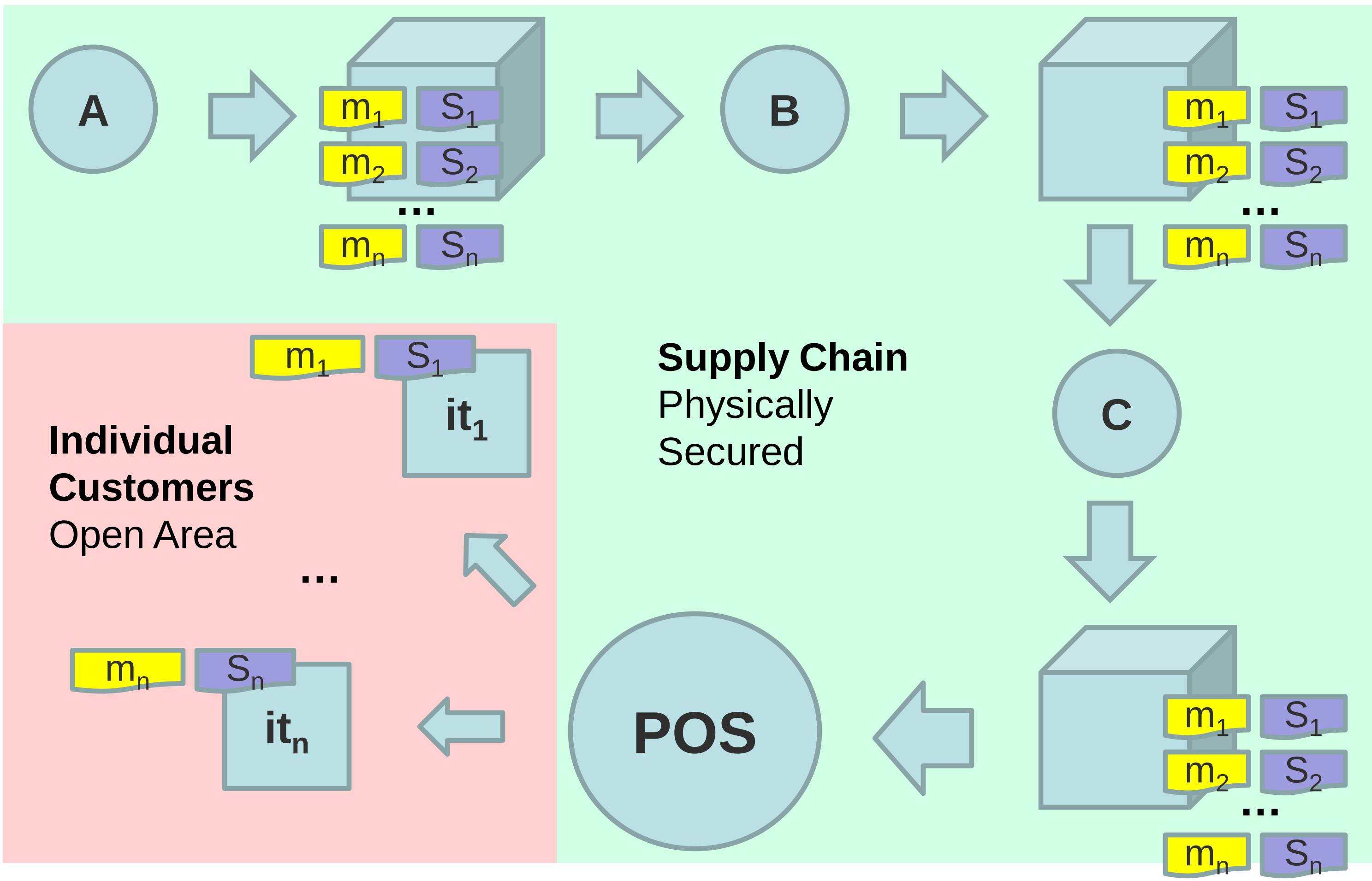
  - Consumer items are typically produced in large quantities
  - Large entities ("cases") are divided into smaller aggregates on their way to the retailers
  - Clients are likely to possess only a small part of one and the same case

Studienstiftung des deutschen Volkes

Secret sharing across space (1)

- General idea and terminology used
  - Deploying secret keys on RFID tags themselves, thus avoiding a supplementary (not too) secure channel
  - Use of secret sharing techniques, e. g. Shamir's [Sha79]
  - All tags in a case are encrypted using the same encryption key κ
  - Key κ is shared among **n** individual tags $T = \{\tau_1, \ldots, \tau_n\}$ using a (**k,n**)-PSS scheme
  - Content of each tag: $v_i = (E_\kappa[m_i], S_i)$
  - Possession of a critical threshold number **k** of shares $\mathbf{S_i}$ indispensable for recovery of secret key **κ** (and, subsequently, for invoking decryption of data strings $\mathbf{m_i}$)

Secret sharing across space (2)



Studienstiftung des deutschen Volkes

Secret sharing across space (3)

- Main advantages of sharing across space
  - Violation of privacy impossible (prevention of "on-the-road" scanning of individual customers)
  - No need for killing tags at the point of sale (POS)
  - Security is ensured "implicitly" by hardness of cryptographic problems
  - No further organizational precautions required
  - Security against loss or damage of a certain number **n-k** of tags used within a case
  - No extra tag per case (in addition to individual item tags) needed

Secret sharing across space (4)

- Problems and restrictions:
  - No security in the face of tracking attacks
    - Using encrypted values as unique identifiers
  - Space restrictions on RFID tags
    - Traditional PSS schemes impose $l(S_i) \geq l(\kappa) = 128bit$
    - Capacity of EPC tags usually is no more than 16 bit
- Imperfect sharing schemes as a trade-off
  - Length of shares is below that of the secret to protect
  - Requirement of all-or-nothing indistinguishability is dropped
  - Partial information gained about the secret is proportional to $\frac{y-t'}{t-t'}$ where $t' \leq y < t$

Studienstiftung des deutschen Volkes

Secret sharing across space (5)

- Related work and important differences
  - CSS schemes (Krawczyk) provide shares with length **independent** of the secret's size
  - "Tiny secret shares" introduce a concept similar to Krawczyk's
  - Use of Error Correcting Codes instead of PSS and IDA schemes
    - Error Correcting Codes = generalization of secret sharing
    - ECCs allow for a smaller size of shares
    - Robustness in the face of (accidentally or deliberately) manipulated shares

Studienstiftung des deutschen Volkes

Secret sharing across space (6)

- Definition of Tiny Secret Share schemata:

  An n-party secret-sharing scheme is a pair of algorithms ***Π = (Share, Recover)*** *that operates over a message space X, where:*

  Share is a probabilistic algorithm that takes input $x \in \mathbb{X}$ and outputs the n-vector $S \xleftarrow{R} \mathsf{Share}(x)$, where $S_i \in \{0,1\}^*$. On invalid input $\hat{x} \notin \mathbb{X}$, Share outputs an n-vector of the special ("undefined") symbol $\perp$.

  Recover is a deterministic algorithm that takes input $S \in (\{0,1\}^* \bigcup \diamondsuit)^n$, where $\diamondsuit$ represents a share that has been erased (or is otherwise unavailable). The output $\mathsf{Recover}(S) \in \mathbb{X} \bigcup \perp$, where $\perp$ is a distinguished value indicating a recovery failure.

Secret sharing across space (7)

- Privacy experiment
  - Only underinformed attacks are considered (for information about overinformed attacks cf [JPP08])
  - Definition of an adversarial game:
    1. The adversary is asked to choose two values
    2. The experiment selects one of them at random and generates a set of shares
    3. The adversary can see individual shares and…
    4. …must produce a guess as to which secret was shared
  - Adversary's advantage is formally defined according to Bellare and Rogaway

Studienstiftung des deutschen Volkes

Secret sharing across space (8)

- Robustness experiment
  - Legitimate users should be able to recover the secret key even if an attacker manipulates some of the shares
  - Definition of an adversarial game:
    1. The adversary chooses a text X
    2. The Share algorithm is invoked
    3. Adversary replaces the share values of a number of players
    4. Adversary is successful, if Recover(X) fails to reconstruct X
  - Adversary's advantage is again formally defined using Bellare's and Rogaway's terminology

Studienstiftung
des deutschen Volkes

Secret sharing across space (9)

- Definition of a (k,n)-TSS scheme

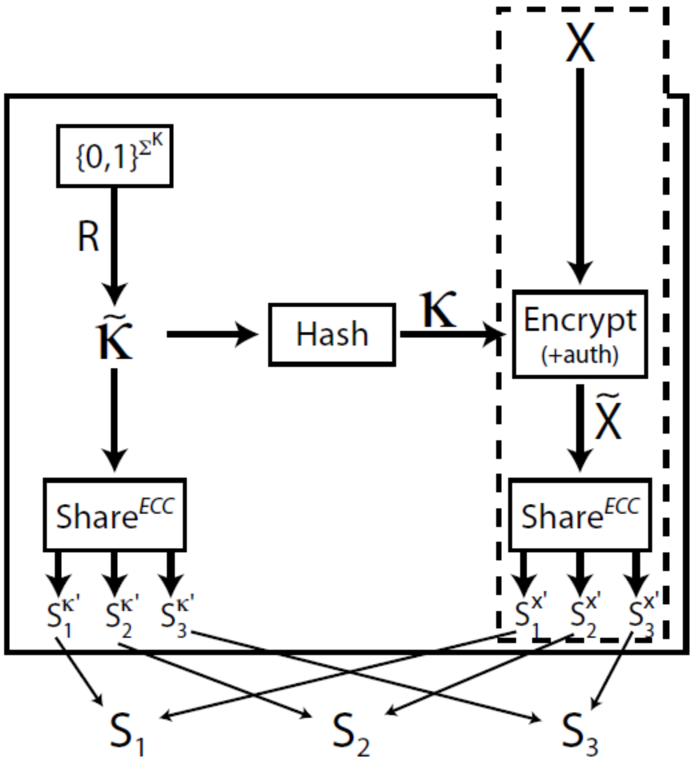
Definition 1 *A (k,n)-TSS scheme is a pair $(\Pi, \mathbb{X})$, such that Π distributes n shares of a secret $x \in \mathbb{X}$, of which any set of k correct shares suffices to recover x. The security of the scheme is characterized by an adversary class $\mathcal{A}$ and the tuple: $(q_u, \varepsilon_u, q_r, \varepsilon_r)$, where an underinformed attacker $A \in \mathcal{A}$ making q_u corrupt queries has $\mathbf{Adv}_A^{ind}[\Pi, \mathbb{X}] \leq \varepsilon_u$; likewise, the pair (q_r, ε_r) applies to robustness attackers. (An extended definition can include overinformed attackers as well; see Appendix A.)*

- Design goal

Development of implementations with ε_u and ε_r as small as possible

Studienstiftung des deutschen Volkes

Secret sharing across space (10)

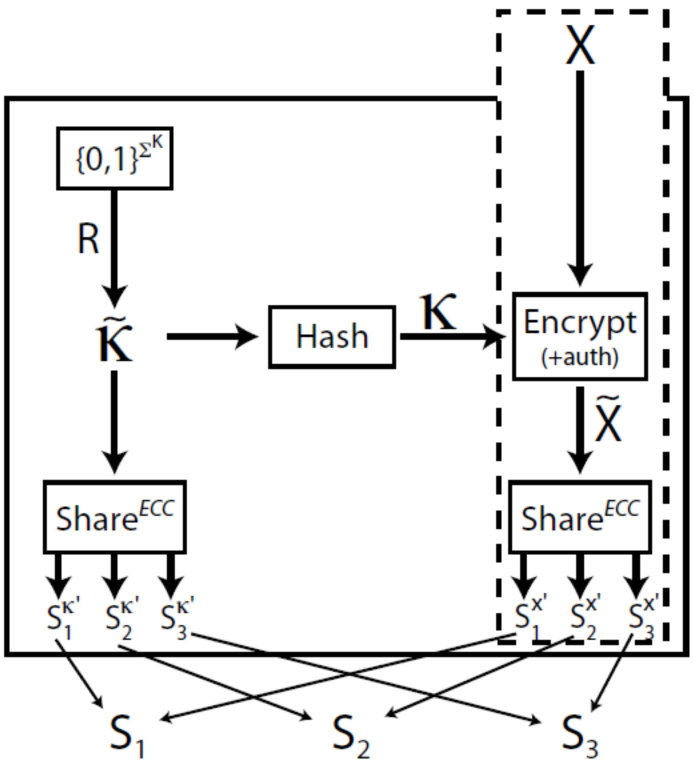


Actions performed by the Share algorithm

1. Input: secret **X**
2. creation of random key $\tilde{K}$
3. $\tilde{K}$ is hashed to K (K is indistinguishable even with $\tilde{K}$ partially compromised)
4. Encrypt **X** using K
5. Create shares of $\tilde{K}$ and $\tilde{X}$ via an ECC

Studienstiftung
des deutschen Volkes

Secret sharing across space (11)



Actions performed by the Recover algorithm

1. Apply the ECC decoding algorithm to recover $\tilde{K}$ and $\tilde{X}$
2. Use $\tilde{K}$ to derive K (use the hash function)
3. Decrypt **X** using K

Remark:

Actions depicted in the dotted box can be omitted provided that key distribution is needed exclusively (as with **RFID tags**!)

Secret sharing across space (12)

- Implementation sketch used by Juels et. al.
  - (15,20)-TSS schemata
  - $GF(2^{16})$, so each share is 16 bits long
  - Random 240-bit $\tilde{K}$ is hashed with SHA-256
  - The first half of the output of SHA-256 is set to be K
  - K is encoded into 20 16-symbols with a (20,15) Reed-Solomon ECC
  - 80 bits left over for the (encoded) tag ID itself
- Real World Parameterization (e. g.)
  - (200,170)-Reed Solomon ECC
  - Appropriate choice of field size (due to memory constraints)

Secret sharing across time (1)

- Why use time-based secret sharing?
  - Schemes developped so far solely take into account one single shipment
  - Most legitimate recipients receive more than one shipment consecutively
  - Time-based secret sharing addresses this characteristic
  - Example:
    - Alice's trucks hold up to ten cases
    - Each case contains a specific tag
    - Alice selects a window of eleven cases a legitimate middleman must possess to be able to manipulate the tags

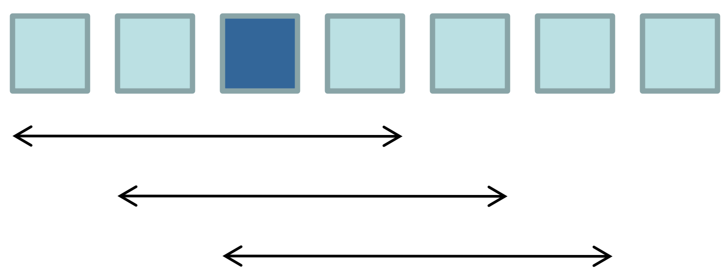
Secret sharing across time (2)

- General idea
  - Alice creates a master secret **κ** (now a write-access key)
  - Any adversary able to recover **κ** is capable of deriving all the write-access PINs valid for the tags in the "window"
  - Case-specific shares may be further distributed on individual item tags (though not necessary)
  - Pre-defining windows is not feasible due to unpredictability of the individual cases Bob is going to receive
  - Defining Sliding-Window Information Secret-Sharing (SWISS) schemes
    - Goal: **Any k** cases in **any** contiguous window of **n** cases suffice to recover **all** the case tags in the window

Studienstiftung des deutschen Volkes

Secret sharing across time (3)

- Naïve idea
  - Generate a key for **every** possible window of size **n** and share each key using a **(k,n)** scheme
  - Each case would have to be equipped with a share for **every** window covering it



n = 4 k = 3

  - Per-case size of shares would grow linearly with **n**
  - Linear growth not acceptable due to space restrictions

Secret sharing across time (4)

- Development of more sophisticated schemes
  - Sequence of shares $S = \{S_0, S_1, \ldots\}$ expanding indefinitely
  - **Each** period has an associated key κ_i
  - Within **any window** of **n** elements, **any k shares S_i** suffice for recovery of all keys
  - Definition of adversarial games similar to SSoS (cf paper)
- Formal definition of (k,n)-SWISS schemata

Definition 2 *We define a (k,n)-SWISS scheme as a pair of algorithms Π as defined above where* Share *produces shares of size μ. The security is characterized by the pair (λ, ε), where (as explained above) k shares are sufficient to reveal λ "nearby" keys for time periods not contained in a window of n shares, and $\mathbf{Adv}_A^{ind-swiss}[\Pi] \leq \varepsilon$.*

The tuple (λ,ε) describes security of a given SWISS implementation
➔ Ideal schema: (λ,ε) = (0,0) with minimal share length μ

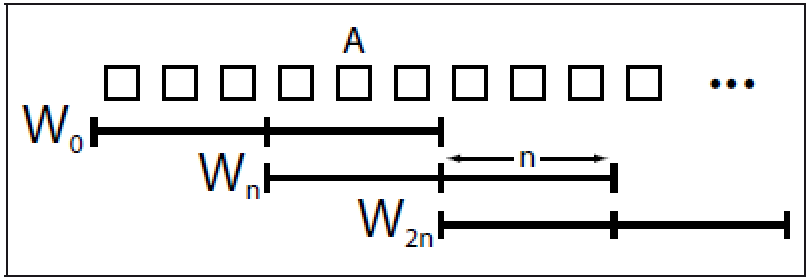
Secret sharing across time (5)

- Main design goals of SWISS schemata
  1. Keys in a window of **n** cases should only be recoverable with access to at least **k** keys in that window (as told before)
  2. Provision of a trade-off between security requirements and memory consumption
  3. Recovery of a case key κ_i requires possession of the very same case (!)
- Innovative approach
  - Share length is designed to be a (small) constant independent of **n** and **k**
  - Goal is achieved using bilinear maps (combination of two multiplicative cyclic groups) – for details cf paper and referred literature

Studienstiftung des deutschen Volkes

Secret sharing across time (6)

- General idea (presented qualitatively)
  - Definition of superwindows of size **2n** overlapping with the previous superwindow by length **n**
  - "Sloppiness" in the resulting access structure: Recovery of secrets in one window gives access to secrets in adjacent windows
  - Each superwindow is given a secret shared using a (k,2n) scheme



  - Access to window secrets (and thus to case secrets) requires recovery of one of the two superwindow secrets
  - **Any k** cases fall into some superwindow of size 2n

Secret sharing across time (7)

- Some implementation details
  - Each time period is covered by two superwindows
  - Hence, each share **S_i** consists of two sub-shares, one for each superwindow
  - Each share contains a supplementary random nonce **r_i** to prevent adversaries from accessing tag secrets for cases they don't possess

$$S_i = \{s_i^{\ell n}, s_i^{(\ell+1)n}, r_i\}$$

  - Individual case keys are defined as follows (taking into account the window secret **i** belongs to and the individual random nonce **r_i**):

$$\kappa_i = h(r_i, \omega_{kn})$$

Studienstiftung des deutschen Volkes

Secret sharing across time (8)

- Generic SWISS families
  - Provide a more flexible implementation of SWISS schemes
  - λ (security parameter) is traded off against memory needed
  - Superwindows are divided into ψ+1 subwindows
- Real world implementation
  - For 1 million shares: 10,000 windows of n = 100 shares each
  - k = 20 ➔ resulting shares will be 384 bits in size
  - Use of TSS schemes provides additional advantages

Outlook on future work

- Future work
  - Creation of sharing schemes based upon the entire history of transaction between sender and receiver