Localised Multisecret Sharing

Thalia M. Laing
Thalia.Laing.2012@live.rhul.ac.uk

Keith M. Martin,
keith.martin@rhul.ac.uk

Information Security Group,
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, UK.

Maura B. Paterson,
m.paterson@bbk.ac.uk
Economics, Mathematics and Statistics,
Birkbeck, University of London,
Malet Street, London WC1E 7HX, UK.

Douglas R. Stinson*,
dstinson@uwaterloo.ca
David R. Cheriton School of Computer Science,
University of Waterloo,
Waterloo, Ontario, Canada N2L 3G1.

September 22, 2015

Abstract

A localised multisecret sharing scheme is a multisecret sharing scheme for an ordered set of players in which players in the smallest sets who are authorised to access secrets are close together in the underlying ordering. We define threshold versions of localised multisecret sharing schemes, we provide lower bounds on the share size of perfect localised multisecret sharing schemes in an information theoretic setting, and we give explicit constructions of schemes to show that these bounds are tight. We then analyse a range of approaches to relaxing the model that provide trade-offs between the share size and the level of security guarantees provided by the scheme, in order to permit the construction of schemes with smaller shares. We show how these techniques can be used in the context of an application to key distribution for RFID-based supply-chain management motivated by the proposal of Juels, Pappu and Parno from USENIX 2008.

1 Introduction

A *secret sharing scheme* is a means of distributing some information (*shares*) to a set of players so that *authorised* subsets of players can recover a unique secret, whereas the shares belonging to *unauthorised* subsets do not reveal any information about the secret. For example, a $(t; n)$-threshold scheme involves a set of n players, with the authorised subsets being all sets of t or more players and the unauthorised subsets being all sets of $t-1$ or fewer players. Such schemes were proposed independently by Blakley and Shamir in 1979 [2, 18]. More generally, given a set $\mathcal{U}$ of players we can define an *access structure* Γ to be the collection of all authorised subsets of $\mathcal{U}$, with all other subsets of $\mathcal{U}$ being unauthorised. We require Γ to be *monotone*, in the sense that for $A, B \subseteq \mathcal{U}$ with $A \subseteq B$ then if $A \in \Gamma$ we have $B \in \Gamma$ also. A *multisecret sharing scheme* is a generalisation of secret sharing in which several secrets are shared according to different access structures on the same set of participants [3, 4, 7, 9, 14]. In this paper we consider a

*D.R. Stinson's research is supported by NSERC discovery grant 203114-11

specific class of multisecret sharing schemes that are suited to an application in RFID security proposed by Juels, Pappu and Parno in USENIX 2008 [10]. We motivate the definition of these schemes initially through the following toy example:

Scenario 1 (Distributed signing key for a committee with changing membership)**.** *A learned society is lead by a committee with seven members. The members each serve a seven year term on the committee; each year one member leaves the committee and a new member is elected to replace them. Every year the society holds a conference in a different city and committee meetings occur at these conferences. A meeting is deemed quorate as long as at least three committee members are present.*

The society wishes to distribute a signing key among the committee members that allows them to sign the reports of their meetings at which at least three members are present. The key will need to change each year to reflect the changed membership of the committee. However they wish to avoid the need to change the members' shares, since the shares are handed out at the committee meetings, but not every member attends each meeting. Furthermore, the shares belonging to any members who have left the committee should not reveal any information about the current value of the committee's signing key.

The requirements of Scenario 1 lead us to the following definition:

Definition 1.1 (Localised threshold multisecret sharing scheme)**.** Let $\mathcal{U} = u_0, u_1, u_2, \ldots$ be an ordered set of players. A *window of length* n consists of a set of n consecutive players, and we denote by W_i the window $\{u_i, u_{i+1}, \ldots, u_{i+n-1}\}$. To each window W_i we assign a secret k_i from some finite secret space K. Let S be a finite set of *shares*. A scheme that associates a share $s_i \in S$ to each player u_i is a *localised multisecret sharing scheme with window length* n *and threshold* t (denoted $(t; n)$-LMSS) if it satisfies the following properties:

- for any $i = 0, 1, 2, \ldots$ the set of shares associated with the players in a set $U \subseteq \mathcal{U}$ allow the secret k_i to be recovered uniquely whenever $|U \cap W_i| \geq t$;
- for any $i = 0, 1, 2, \ldots$ if $U \subseteq \mathcal{U}$ is a set of players with the property that $|U \cap W_i| < t$ then the shares associated with the players in U reveal no information about k_i.

Example 1.1. The use of a $(3; 7)$-LMSS to distribute their signing keys would allow the committee in Scenario 1 to satisfy their requirements: the first seven committee members are the players $u_0, u_1, \ldots, u_6$, and the remaining committee members are ordered by the year in which they join the committee. Each player u_i is given a share s_i. As illustrated in Figure 1, window

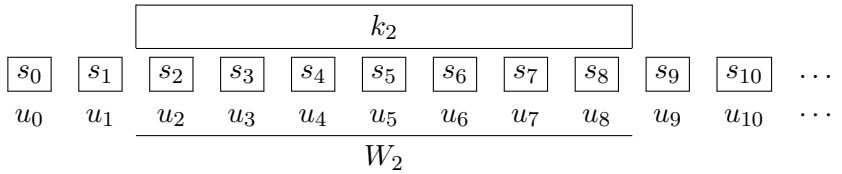

Figure 1: Depiction of window W_2 in a $(3; 7)$-LMSS.

W_2 contains players u_2, u_3, u_4, u_5, u_6, u_7 and u_8, which correspond to the members who are in the committee during year 2. Any three or more of these members can combine their shares to recover the signing key k_2 for year 2. For example, the set of members $\{u_3, u_6, u_8\}$ is an authorised set for year 2, so shares s_3, s_6 and s_8 can be used to reconstruct k_2. On the other hand, the set $\{u_7, u_8, u_9, u_{10}\}$ only contains two members from within window W_2 and hence

the set of shares $\{s_7, s_8, s_9, s_{10}\}$ do not reveal any information about k_2. (They would, however, allow k_3, k_4, k_5, k_6, k_7 or k_8 to be recovered.)

The key feature of an LMSS is the fact that the set of players is ordered and every minimal authorised set of players is contained within a window of n consecutive players for some n. The aim of this paper is to analyse the security of such schemes and provide efficient techniques for their construction. In Section 2 of this paper we provide some necessary background details on secret sharing and multisecret sharing. In Section 3 we provide theoretical results on the security properties of LMSS including bounds on the share sizes, and give constructions of schemes that meet these bounds. In Section 4 we explore how the definition of a LMSS can be relaxed in order to permit schemes with smaller share sizes, and in Section 5 we show how these ideas can be applied in a natural way in designing a scheme suitable for an application in RFID-based supply chain management motivated by a proposal of Juels, Pappu and Parno [10].

2 Background

Here we summarise some results and techniques from the literature that we use later in this paper.

2.1 Secret Sharing

The earliest proposed examples of secret sharing schemes were for $(t; n)$-threshold schemes. The following construction is due to Shamir and yields a $(t; n)$-threshold scheme for any positive integers t, n with $t \leq n$:

Construction 1 (Shamir's Secret Sharing Scheme [18])**.** *Let $\mathcal{U}$ be a set of n players, and let $p > n$ be a prime. For a given secret $k \in \mathbb{Z}_p$, select $t-1$ further values r_1, r_2, r_{t-1} uniformly at random from $\mathbb{Z}_p$, and let $f \in \mathbb{Z}_p[x]$ be the polynomial defined by*

$$f(x) = r_{t-1}x^{t-1} + r_{t-2}x^{t-2} + \cdots + r_1 x + k.$$

We identify each player with a unique nonzero element of $\mathbb{Z}_p$, and to player i we assign the share $f(i)$.

Any set of t or more players can perform polynomial interpolation on their shares in order to recover the polynomial f and hence determine the secret k. However, for any set of $t-1$ or fewer players, and for any element $k' \in \mathbb{Z}_p$, there exists a polynomial of degree at most $t-1$ consistent with their shares and having constant term k'. Thus the shares of an unauthorised set of players yield no information about the true value of k. Note that it is possible to replace $\mathbb{Z}_p$ by the finite field $\mathrm{GF}(q)$ for any prime power $q > n$, and that a slight adjustment can make the scheme work for $q \geq n$.

The properties of secret sharing schemes are often described using information theoretic notation. Let $\mathbf{K}$ denote the discrete random variable corresponding to the choice of secret, and let $\mathbf{A}$ denote the discrete random variable corresponding to the set of shares given to the players in the set $A \subseteq \mathcal{U}$. For a secret sharing scheme with access structure Γ we require $\mathrm{H}(\mathbf{K} \mid \mathbf{A}) = 0$ for any authorised set $A \in \Gamma$, and $\mathrm{H}(\mathbf{K} \mid \mathbf{B}) = \mathrm{H}(\mathbf{K})$ for any unauthorised set $B \notin \Gamma$. A secret sharing scheme satisfying the second of these properties is said to be *perfect*; Shamir's secret sharing scheme is an example of a perfect scheme. It is well known that for any perfect secret sharing scheme, then $\mathrm{H}(\mathbf{S_i}) \geq \mathrm{H}(\mathbf{K})$, where $\mathbf{S_i}$ is the discrete random variable corresponding to

the share belonging to player u_i. Informally speaking, the size of each share is at least the size of the secret for any perfect scheme.

One way to allow the construction of schemes with smaller shares is to relax the requirement for the scheme to be perfect. For example, a $(t_1, t_2; n)$-*ramp scheme* is a secret sharing scheme in which any t_2 players can use their shares to uniquely recover the secret, whereas the set of shares belonging to any set of t_1 or fewer players reveals no information about the secret. Sets of players of sizes greater than t_1 but smaller than t_2 may learn partial information about the secret, hence ramp schemes are not perfect. The average entropy of a player's share in a $(t_1, t_2; n)$-ramp scheme is known to be at least $\frac{\log_2 |K|}{t_2 - t_1}$ [12]. The following construction is based on Construction 1, and achieves $\mathrm{H}(\mathbf{S_i}) = \frac{\mathrm{H}(\mathbf{K})}{t_2 - t_1}$ for each player u_i; it is due to McEliece and Sarwate [15].

Construction 2 ($(t_1, t_2; n)$-ramp scheme [15])**.** *Set $s = t_2 - t_1$. Let $\mathcal{U}$ be a set of $n - s + 1$ players, and let $p > n$ be a prime. The secret for this scheme is an element $\mathbf{k} = (k_1, k_2, \ldots, k_s) \in \mathbb{Z}_p^s$. It is shared by selecting a polynomial f uniformly from the set of all polynomials in $\mathbb{Z}_p[s]$ that satisfy $f(0) = k_1, f(1) = k_2, \ldots, f(s-1) = k_s$. We identify each player with a unique nonzero element of $\mathbb{Z}_p \setminus \{0, 1, \ldots, s-1\}$, and to player i we assign the share $f(i)$.*

As before, any set of t_2 players can perform interpolation to recover f, which enables them to recover the entire secret. In addition, it can be shown that any set of t_1 or fewer players learns no information about the secret.

2.2 Multisecret Sharing

Many authors have studied a generalisation of secret sharing in which several secrets are shared according to different access structures on the same set of participants [3, 4, 7, 9, 14]. There are various equivalent ways of defining security in such a setting; the following is due to Herranz, Ruiz and Sáez [7]:

Definition 2.1. Let $\mathcal{U}$ be a set of players, and for $j = 1, 2, \ldots, m$ let Γ_j be a monotone access structure on $\mathcal{U}$. Let Q be the set of possible secrets, and suppose that secrets $k_1, k_2, \ldots, k_m$ are chosen from Q. A scheme that allocates to each player u_i in $\mathcal{U}$ a share s_i from some set S of potential shares is a *weakly information-theoretic secure multi-secret sharing scheme* if it satisfies the following properties:

- *(correctness)* for $i = 1, 2, \ldots, m$ and for any set $A \subset \mathcal{U}$ of players we have that if $A \in \Gamma_j$ then the shares of the players in A can be used to uniquely recover secret k_j. In terms of entropy, $\mathrm{H}(\mathbf{K_j} \mid \mathbf{A}) = 0$ whenever $A \in \Gamma_j$.
- *(weak information-theoretic security)* if $A \notin \Gamma_j$ then the shares of the players in A reveal no information about k_j. That is, $\mathrm{H}(\mathbf{K_j} \mid \mathbf{A}) = \mathrm{H}(\mathbf{K_j})$.

If, in addition, any set $A \notin \Gamma_j$ together with a set T of secrets with $k_j \notin T$ reveals no information about k_j that is not already revealed by T alone (*i.e.* if $\mathrm{H}(\mathbf{K_j} \mid \mathbf{A}, \mathbf{T}) = \mathrm{H}(\mathbf{K_j} \mid \mathbf{T})$) then the scheme is said to have *strong information-theoretic security.*

A $(k; n)$-LMSS is a special case of a multi-secret sharing scheme; the version presented in Definition 1.1 corresponds to the case of a weakly information theoretically secure scheme. Our analysis of these schemes in Section 3 will require the following theorem of Herranz, Ruiz and Sáez:

Theorem 2.1 ([7])**.** *Let $\Gamma_1, \ldots, \Gamma_\ell$ be ℓ access structures on $\mathcal{U}$ and consider the player $u_i \in \mathcal{U}$. Assume there exist subsets of players $B_1 \subset B_2 \subset \ldots \subset B_\ell \subset \mathcal{U} \setminus \{u_i\}$ satisfying, for all $j = 1, 2, \ldots, \ell$, the following three conditions:*

1. $B_j \in \Gamma_{j-1}$ *whenever* $j > 1$;
2. $B_j \notin \Gamma_j$;
3. $B_j \cup \{u_i\} \in \Gamma_j$.

Then, for any weakly-information theoretic secure multi-secret sharing scheme for $\Gamma_1, \Gamma_2, \ldots, \Gamma_\ell$, *it holds that* $\mathrm{H}(\boldsymbol{S_i}) \geq \sum_{i=1}^{\ell} \mathrm{H}(\boldsymbol{K_i})$.

3 Bounds and constructions for LMSSs

A $(t;n)$-LMSS can be regarded as a weakly information-theoretic secure multisecret sharing scheme with access structures $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$, where Γ_i is a $(t;n)$-threshold access structure for the set of players within window W_i. We note that it is possible to construct such a scheme by deploying a $(t;n)$-threshold scheme in each window, as observed in [10] (the same approach has been mentioned previously in the literature as a way to construct multisecret sharing schemes for various other combinations of threshold access structures, *e.g.* [3]).

Construction 3 (Trivial $(t;n)$-LMSS [10])**.** *Let* $\mathcal{U} = u_0, u_1, u_2, \ldots$ *be an ordered set of players, and denote by* W_i *the window* $\{u_i, u_{i+1}, \ldots, u_{i+n-1}\}$. *Let* p *be a prime with* $p > n$. *For each window* W_i *we share a secret* $k_i \in \mathbb{Z}_p$ *among the* n *players in* W_i *using a* $(t;n)$*-threshold scheme (Construction 1). This is done independently for each window; a player* u_j *is thus assigned shares corresponding to each window that contains it, i.e. windows* $W_{j-n+1}, W_{j-n+2}, \ldots, W_j$. *(The secrets may or may not be independent, but the randomness used in each of the threshold schemes is chosen independently.))*

It is straightforward to see that Construction 3 gives rise to a $(t;n)$-LMSS in which the total share size for each player is $n \log p$. In the case where $k < n$ this turns out to be optimal, as shown by the following theorem:

Theorem 3.1. *Let* $\mathcal{U} = u_0, u_1, u_2, \ldots$ *be an ordered set of players. Suppose* $t < n$ *and let* Σ *be a weakly secure* $(t;n)$-LMSS *that associates a share* $s_i \in Q$ *to each player* u_i. *Then for any* $j \geq n-1$ *we have that*

$$\mathrm{H}(\boldsymbol{S_i}) \geq \sum_{i=j-n+1}^{j} \mathrm{H}(\mathbf{K_i}).$$

Proof. For ease of notation we prove the result for $j = n-1$, but the proof applies analogously to any $j \geq n-1$. Consider the sequence of players $\mathcal{P} = [u_0, u_1, \ldots, u_{2n-2}]$. Restricting Σ to these players yields a weak information-theoretic secure multi-secret sharing scheme on $\mathcal{P}$ where for $i = 0, \ldots n-1$ the secret k_i is shared according to the access structure $\Gamma_i = \{S \subset W_i \mid |S| \geq t\}$.

For $r = 1, 2, \ldots, n$, let B_r consist of the first $t - 2 + r$ elements of $\mathcal{P} \setminus \{u_{n-1}\}$. Then the sets B_i satisfy the conditions for Theorem 2.1, and so we have that $\mathrm{H}(\boldsymbol{S_{n-1}}) \geq \sum_{i=0}^{n-1} \mathrm{H}(\boldsymbol{K_i})$ as required. □

If the secrets and shares are all uniformly distributed, we obtain the result that the size of the secret given to player P_j with $j \geq n-1$ in a $(t;n)$-LMSS with $t < n$ is at least $n \log |K|$, which implies that Construction 3 is optimal. (We address the case $t = n$ in Section 3.1.) The fact that there do not exist constructions for a $(t;n)$-LMSS with share sizes shorter than those of Construction 3 means that these schemes are not suitable for applications that require small shares. In Section 4 we consider various approaches to relaxing the security definitions for these schemes in a controlled manner so as to allow more efficient constructions. On the positive side, we observe that if the secrets are generated independently of each other, then the optimal schemes of Construction 3 are in fact strongly information-theoretic secure.

3.1 The case $t = n$

Interestingly, the restrictions of Theorem 3.1 do not apply in the case where $t = n$, as the following construction demonstrates.

Construction 4. *Let $\mathcal{U} = u_0, u_1, u_2, \ldots$ be an ordered set of players, and denote by W_i the window $\{u_i, u_{i+1}, \ldots, u_{i+n-1}\}$. Suppose that for $i \geq 0$ a secret k_i is generated uniformly at random from the set $\mathbb{Z}_2^\lambda$, and that this is done independently for each i. For $i = 0, 1, \ldots, n-2$ we assign a share s_i to player u_i by generating s_i uniformly at random from $\mathbb{Z}_2^\lambda$. For $i \geq n-1$ we set $s_i = k_{i-n+1} \oplus s_{i-n+1} \oplus s_{i-n+2} \oplus \cdots \oplus s_{i-1}$.*

Theorem 3.2. *Construction 4 results in an $(n; n)$-LMSS that has weak information-theoretic security and optimal share size, but which does not possess strong information-theoretic security.*

Proof. In an $(n; n)$-LMSS, the access structure Γ_j has a single minimal authorised subset A, which is the set consisting of all n players in the window W_j:

$$A = \{u_j, u_{j+1}, ..., u_{j+n-1}\}.$$

Note that the secret k_j can be recovered from the shares belonging to players in A by calculating $k_j = s_j \oplus s_{j+1} \oplus \cdots \oplus s_{i+n-2} \oplus s_{j+n-1}$.

For the access structure Γ_j, the maximal unauthorised subsets have the form

$$B_{\hat{i}} = \{u_0, u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_{j+n-1}, \ldots\},$$

for some i with $j \leq i \leq j + n - 1$.

Suppose an attacker who wishes to determine k_j possesses all the shares in $B_{\hat{i}}$.

By definition of Construction 4,

$$s_{j+n-1} = k_j \oplus s_j \oplus s_{j+1} \oplus \cdots \oplus s_{i-1} \oplus s_i \oplus s_{i+1} \oplus \cdots \oplus s_{j+n-2},$$

hence

$$k_j = s_j \oplus s_{j+1} \oplus \cdots \oplus s_{i-1} \oplus s_i \oplus s_{i+1} \oplus \cdots \oplus s_{j+n-2} \oplus s_{j+n-1}. \tag{1}$$

The adversary knows all terms on the right hand side of (1) apart from the share s_i, which we can express as

$$s_i = k_{i-n+1} \oplus s_{i-n+1} \oplus s_{i-n+2} \oplus \cdots \oplus s_{i-1}. \tag{2}$$

Strong security We show the construction does not have strong information-theoretic security. In the strong security setting, in addition to the shares possessed by the players in $B_{\hat{i}}$, we assume the adversary also knows all secrets other than k_j. In the case where $i \neq j + n - 1$, this includes the secret k_{i-n+1}, and so the adversary is able to determine s_i using (2), and hence recover k_j using (1).

When $i = j + n - 1$ (that is, when the missing share is the last one in the window) the adversary does not possess k_{i-n+1}. However, we observe that:

$$s_{j+n} = k_{j+1} \oplus s_{j+1} \oplus s_{j+1} \oplus ... \oplus s_{j+n-2} \oplus s_{j+n-1}.$$

The adversary possesses all terms in this expression other than $s_{j+n-1} = s_i$, hence they can use it to recover s_i and thence k_j as before.

Weak security In the weak information theoretic setting, the adversary does not know any secrets others than those it is able to compute using the shares in its possession. Consider the following system of equations:

$$\begin{aligned} k_{i-n+1} &= s_{i-n+1} \oplus s_{i-n+2} \oplus \cdots \oplus s_{i-1} \oplus s_i, \\ k_{i-n+2} &= s_{i-n+2} \oplus s_{i-n+3} \oplus \cdots \oplus s_i \oplus s_{i+1}, \\ &\vdots \\ k_i &= s_i \oplus s_{i+1} \oplus ... \oplus s_{i+n-2} \oplus s_{i+n-1}. \end{aligned}$$

This is a system of n linear equations. The adversary knows all the values except for the $n+1$ values in the set $S = \{s_i, k_{i-n+1}, k_{i-n+2}, \ldots, k_i\}$. For every possible choice of $k_j \in \mathbb{Z}_2^\lambda$ there is a choice of the values in $S \setminus \{k_j\}$ consistent with the set of shares corresponding to players in B. Hence these shares reveal no new information about the secret k_j, meaning this construction has weak information theoretic security.

Optimality To see that this scheme is optimal with respect to share size, we note that restricting this construction to any window gives an $(n; n)$-threshold scheme for that window in which the size of each share is the same as the size of the corresponding secret. Since for a perfect $(n; n)$-threshold scheme the size of the shares must be at least as large as the size of the secrets, this is optimal.

□

The scheme of Construction 4 thus gives an example of a $(k; n)$-LMSS that is weakly secure, but not strongly secure. We note that it has independently generated secrets, thus demonstrating that having independently generated secrets is a necessary but not sufficient condition to guarantee that a weakly secure scheme will also be strongly secure.

3.2 Time dependent schemes

In many applications such as that of Scenario 1, the secrets $k_0, k_1, \ldots$ of a LMSS have a natural interpretation as a sequence of secrets that change over time. In such a setting it makes sense to consider a security model that is intermediate between the strong and weak models:

Definition 3.1. A (weakly information theoretically secure) $(k; n)$-LMSS is said to have *perfect backward secrecy* if the shares possessed by any set $A \notin \Gamma_j$ together with the set of the first j secrets $T_j = \{k_0, k_1, \ldots, k_{j-1}\}$ reveals no information about the secret k_j other than that already revealed by T_j. In entropy terms, $\mathrm{H}(\mathbf{K_j} \mid \mathbf{A}, \mathbf{T_j}) = \mathrm{H}(\mathbf{K_j} \mid \mathbf{T_j})$.

Perfect backward secrecy ensures that the exposure of past secrets does not affect the security of future secrets. Note that, by definition, a $(t; n)$-LMSS with perfect backward secrecy is also a weakly information-theoretic $(t; n)$-LMSS. Furthermore, a strongly information-theoretic $(t; n)$-LMSS necessarily has perfect backward secrecy. Thus perfect backward secrecy can be seen as an intermediate requirement between weak and strong information-theoretic security. A strongly information-theoretic scheme in fact possesses both perfect backward secrecy and perfect forward secrecy, where compromise of future secrets does not affect the security of past secrets. This is interesting from the point of view of motivating the strong security model, given that Scenario 1 seemed *a priori* only to require weak security.

It is interesting to consider whether the scheme of Construction 4 has this property. The following result shows that in fact it only has quite limited backward secrecy:

Theorem 3.3. *Construction 4 does not have perfect backward secrecy: an adversary possessing the secrets $k_0, k_1 \dots k_{j-1}$ and who also has shares of a maximal set of players $B_i \notin \Gamma_j$ can determine k_j except in the case where $i = j + n - 1$, where no information about k_j is revealed.*

Proof. Consider the proof of Theorem 3.2. In showing that an adversary who has access to all secrets other than k_j can recover k_j in the case where $i \neq j + n - 1$ we in fact only made use of secrets k_ℓ with $\ell < j$. Hence the same argument demonstrates that this construction does not give a backward secure scheme. Interestingly, when $i = j + n - 1$ the adversary in fact learns nothing about k_j in the backward secure setting. To see this, we consider the following set of equations:

$$\begin{aligned} k_{i-n+1} &= s_{i-n+1} \oplus s_{i-n+2} \oplus \cdots \oplus s_{i-1} \oplus s_i, \\ k_{i-n+2} &= s_{i-n+2} \oplus s_{i-n+3} \oplus \cdots \oplus s_i \oplus s_{i+1}, \\ &\vdots \\ k_i &= s_i \oplus s_{i+1} \oplus \cdots \oplus s_{i+n-2} \oplus s_{i+n-1}. \end{aligned}$$

This is a set of n linear equations with $n + 1$ unknowns, namely $\{k_{i-n+1}, k_{i-n+2}, \dots, k_i, s_i\}$. For every possible secret k_{i-n+1} there exist a choice for the remaining elements of this set that is consistent with the view of the adversary. Therefore, the construction does ensure that an adversary lacking share s_{i-n+1} learns no information about k_{i-n+1}. □

4 Relaxing security requirements in order to construct more efficient schemes

The bounds on share sizes implied by Theorem 3.1 mean that in order to construct a more efficient $(k; n)$-LMSS it is necessary to relax the security definition. There are various ways in which this could be done. The *SWISS schemes* proposed by Juels *et al.* are one example [10]; we discuss some limitations of these schemes in Section 5. In this Section we consider systematically a range of techniques that can be applied while still working in the setting of information-theoretic security. Recall that the essential aims of a $(k; n)$-LMSS are to ensure that any k suitably close users are able to recover a secret, and that each secret should only be accessible to players within a bounded window. The techniques we consider here allow us to maintain these goals, while relaxing the strict requirements of Definition 1.1 in ways that give us a well understood trade-off between the security compromises and the resulting efficiency gains.

4.1 Shifting to a nonperfect model of secret sharing

Section 2.1 indicated that the shares of a perfect $(t; n)$-threshold scheme have to be at least as large as the secret, but that share sizes could be reduced by the use of a $(t_1, t_2; n)$-ramp scheme, with the increased efficiency being traded against the relaxation of the security in that sets of players of sizes between t_1 and t_2 can now gain partial information about the secret. The exact same technique can be applied in the context of a $(t; n)$-LMSS, by replacing the use of Shamir's secret sharing scheme in Construction 3 with the $(t_1, t_2; n)$-ramp scheme of Construction 2. Reducing the sizes of sets of players that are excluded from learning any information about the secret from $t_2 - 1$ to t_1 in this manner allows us to decrease the size of the shares by a factor of $t_2 - t_1$.

4.2 Changing the access structures

One consequence of Definition 1.1 is that the functionality (in terms of which keys a given player can contribute to recovering) is inextricably tied directly to the security (in terms of the sizes of the windows of players that can contribute to recovering a particular key.) Specifically, if we wish to consider windows of length n, then according to this definition each player is necessarily a participant in n distinct windows. As this connects directly to the storage overheads for each player, this restriction limits the scope for improving the efficiency of such schemes. It would be desirable to have greater flexibility in varying the parameters of a scheme. One way to achieve this is to decouple the access structure (*i.e.*, the definition of which sets of players are authorised to access keys) from the pattern of key distribution (*i.e.*, which keys are able to be accessed by particular authorised sets.) In this section we will set up a framework for analysing this more general setting, and explore the resulting consequences in terms of security and practicality. In Section 5 we will see how these more general schemes can be deployed effectively in a practical setting.

The access structure in isolation: As previously, we consider an ordered set of players denoted by $\mathcal{U} = u_0, u_1, u_2, \ldots$. We define an access structure Γ on $\mathcal{U}$ by specifying that the authorised sets in Γ are all those subsets of $\mathcal{U}$ that contain a subset of the form $S = \{u_{i_1}, u_{i_2}, \ldots, u_{i_t}\}$ where $i_1 < i_2 < \cdots < i_t$ and $i_t - i_1 \leq n - 1$. That is, any subset of $\mathcal{U}$ that contains t or more players from within a window of n consecutive players is authorised. We think of the authorised sets in Γ as being those that have the right to reconstruct at least one key.

Key windows: The defining property of a localised secret sharing scheme is that we want any given key to be accessible by sufficiently large sets of players that are suitably close. For a given key k we suppose there is a specific *key window* W^k consisting of the players $u_i, u_{i+1}, \ldots u_{i+\ell-1}$ that we think of as having the potential to be involved in recovering k. Note that unlike in Section 3, we no longer require $\ell = n$ but we can also allow $\ell \geq n$.

The effect of Γ within a key window: We now consider the restriction of the access structure Γ to the window W^k. This gives us an access structure Γ^k on the players in W^k whose authorised sets are all those of the form $A \in \Gamma$ with $A \subseteq W^k$. This is illustrated in Figure 2 for the case $\ell = 4$ and $n = k = 2$.

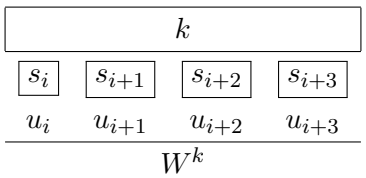

Figure 2: For $\ell = 4$ and $n = k = 2$, any pair of adjacent players are authorised to recover the secret.

For $\ell > n$ the access structure Γ^k is no longer a threshold access structure. It is possible to construct a secret sharing scheme assigning shares to the players in W^k that enable authorised sets in Γ^k to recover k while preventing the shares belonging to any set $S \notin \Gamma^k$ from gaining any information about k [8]. However, in general this may require the share sizes to be larger than the size of k. For example, in the case where $\ell = 4$ and $n = t = 2$ (as depicted in Figure 2),

Capocelli *et al.* show that it is necessary for the largest share to be at least 50% larger than the size of the secret [5].

One approach to avoiding this issue (as suggested in [10]) is simply to use a $(t;\ell)$-threshold scheme to share k among the players in W^k, since this allows shares that are no larger than the size of the secret (and permits a straightforward trade-off between security and share size through the use of a suitable ramp scheme if desired). However, in the case where ℓ is significantly larger than n (the schemes proposed in [10] have $\ell \geq 2n$) this results in many subsets of W^k that are not authorised having shares that allow them to recover the secret, namely any set of t shares $\{s_{i_1}, s_{i-2}, \ldots, s_{i_t}\}$ with $i_1 < i_2 < \cdots < i_t$ and $i_t - i_1 > n$. This can be substantially mitigated at no extra cost by the use of the following construction:

Construction 5. *We generate shares in the secret k for the players in W^k as follows:*

- *We share k using a $(t;n)$-threshold scheme and assign the resulting shares $s_1, s_2, \ldots, s_n$ to the first n players in the window.*
- *We then assign $s_1, s_2, \ldots, s_n$ in turn to the next n players in turn and so on, cycling through the shares as necessary throughout the rest of the window.*

This construction ensures that the shares possessed by any n consecutive players are precisely those of a $(t;n)$-threshold scheme, and so the shares of any subset of t or more of those n consecutive players can recover the secret, as required by Γ^k. There is potential for a small saving in the size of the shares relative to using a $(t;\ell)$-threshold scheme since it is possible to use a field of size n rather than ℓ. However the main advantage of this construction is in reducing the number of sets $S \notin \Gamma^k$ that can recover k. For a $(t;\ell)$-threshold scheme, there are $\binom{\ell}{t}$ subsets of size t that can recover the secret, whereas Γ^k contains only $(\ell - n)\binom{n-1}{t-1} + \binom{n}{t}$ authorised subsets of size t (there are $\binom{n}{t}$ ways of choosing such a subset from among the last n players in the window; for subsets not wholly contained within the last n players there are $\ell - n$ possible choices for the first player in the subset and $\binom{n-1}{t-1}$ ways to choose the rest of the subset from the $n-1$ subsequent players.). For example, if $\ell = 4$ and $n = t = 2$ then the threshold scheme has $\binom{4}{2} = 6$ pairs of shares that can recover the secret, whereas Γ^k only has $2\binom{1}{1} + \binom{2}{2} = 3$ authorised pairs. This implies that half of the pairs of players enabled by the threshold scheme to access the secret are not in Γ^k. For the scheme given in Construction 5, on the other hand, in the case where $\ell = \lambda n$ there are $\lambda^t\binom{n}{t}$ subsets of size t that can recover the secret. For $\ell = 4$ and $n = t = 2$ this gives $2^2\binom{2}{2} = 4$ pairs that can recover the secret, so only a quarter of these are not in Γ^k.

We note that Construction 5 can also be instantiated with a ramp scheme in place of the threshold scheme if desired.

4.3 Staggering key windows

If we were to fix $\ell = n$ and require a new key window of length n starting with player u_i for every $i = 1, 2, \ldots$ then we would recover Definition 1.1. However, by allowing distinct values of ℓ and n, and allowing more flexibility in the distribution of the key windows we can obtain a family of schemes that have the potential for much greater flexibility in tailoring their properties to suit our application requirements. One way to do this is to introduce a parameter d that describes the offset between consecutive key windows, so that key k_0 is associated with the window $W^{k_0} = \{u_0, u_1, u_2, \ldots, u_{\ell-1}\}$, key k_1 is associated with the window $W^{k_1} = \{u_d, u_{d+1}, \ldots, u_{d+\ell-1}\}$, and so on. In general, key k_i is associated with the window $W^{k_i} = \{u_{id}, u_{id+1}, \ldots, u_{id+\ell-1}\}$.

The following lemmas describe basic properties of the scheme that arises from staggering the key windows in this fashion.

Lemma 4.1. *An authorised set $A = \{u_{i_1}, u_{i_2}, \ldots, u_{i_t}\}$ with $i_t - i_1 = c$ for some $c \leq n-1$ can reconstruct $\lfloor \frac{\ell - c}{d} \rfloor$ or $\lceil \frac{\ell - c}{d} \rceil$ secrets.*

Proof. The shares corresponding to the players in set A allow them to recover the secret for any window W with $A \subseteq W$. A window of length ℓ contains A if it starts with a player between $u_{i_{t-\ell}}$ and u_{i_1}. This is a range of $u_{i-1} - u_{i_t-\ell} = \ell - (u_{i_t} - u_{i_1}) = \ell - c$ possible starting points. If the windows have offset d then for any value of i_t at least $\lfloor \frac{\ell - c}{d} \rfloor$ windows, and up to $\lceil \frac{\ell - c}{d} \rceil$ windows will start in this range. □

The share storage requirements for any individual player are given by the following lemma:

Lemma 4.2. *Any single player is associated with either $\lfloor \frac{\ell}{d} \rfloor$ or $\lceil \frac{\ell}{d} \rceil$ shares.*

The proof is a direct analogue of that of Lemma 4.1.

Example 4.1. Suppose we take $\ell = 2n$ and $d = n$. Then every player is associated with shares from two distinct key windows, and any authorised set is able to recover either one or two distinct window keys.

4.4 Combining techniques

Combining all the techniques discussed in this section gives us the following construction:

Construction 6 (Flexible Localised Multisecret Sharing Scheme $(t_1, t_2; n, \ell, d)$-fLMSS)**.** *Let $\mathcal{U} = u_0, u_1, u_2, \ldots$ be an ordered set of players, and denote by W_i the window $\{u_i, u_{i+1}, \ldots, u_{i+\ell-1}\}$. Let p be a prime with $p > n$. For each window W_i for $i = 0, d, 2d, 3d, \ldots$ we share a secret $\sigma_i \in \mathbb{Z}_p$ among the ℓ players in W_i using Construction 5 implemented with a (t, k, n)-ramp scheme.*

The following properties of this construction follow directly from the earlier results in this section.

Theorem 4.3. *A $(t_1, t_2; n, \ell, d)$-*fLMSS *has the following properties:*

- *(storage) If the secrets are all independent and identically distributed according to the uniform random variable $\mathbf{K}$ then the size of the shares of each tag are $\lceil \frac{\ell}{d} \rceil \frac{\log_2 |K|}{t_2 - t_1}$ bits.*
- *(key recovery) Any set of players $\{u_{i_1}, u_{i_2} \ldots, u_{i_m}\} \subseteq W_i$ for $i = 0, d, 2d, \ldots$ with $m \geq t_2$ and $i_m - i_1 \leq n$ is able to recover the secret k_i.*

Example 4.2. Consider key windows of length $\ell = 150$, with an offset of $d = 40$ between consecutive window keys. Let $n = 100$, $t_2 = 50$ and $t_1 = 30$, so any 50 players $u_{i_1}, u_{i_2}, \ldots, u_{i_{50}}$ with $i_{50} - i_1 \leq 99$ are able to fully construct a common secret, whereas sets of 30 or fewer players learn no information about the secret. Using Lemma 4.1, an authorised set $A = \{u_{i_1}, u_{i_2}, \ldots, i_{i_{50}}\}$ with $i_{50} - i_1 = c$ for some $c \leq 99$ is able to reconstruct $\lfloor \frac{150 - c}{40} \rfloor$ or $\lceil \frac{150 - c}{40} \rceil$ secrets. For example, when $c = 50$ we have

$$\frac{\ell - c}{d} = \frac{150 - 50}{40} = 2.5,$$

so each authorised subset is always able to construct at least two, and potentially up to three, window secrets. To illustrate this, consider the authorised set $A = \{u_{120}, u_{121}, \ldots, u_{169}\}$, which is depicted in Figure 3. The shares in A are capable of reconstructing the secrets W_{40}, W_{80} and W_{120}. This is the maximum possible. Consider a different authorised set

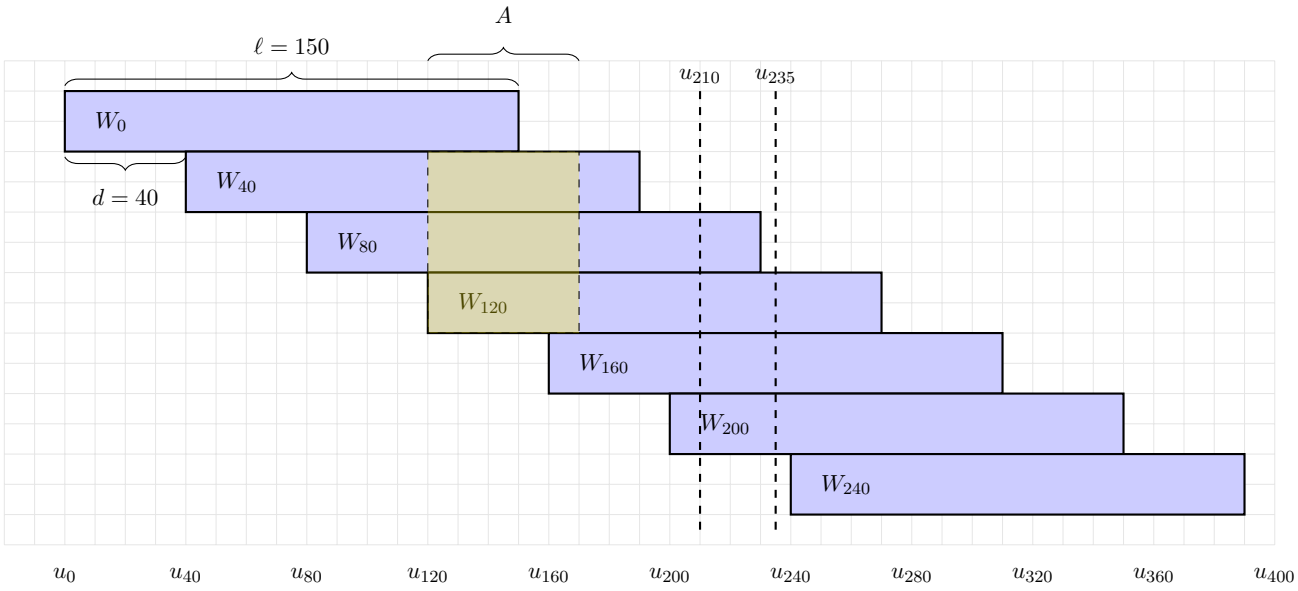

Figure 3: Illustration of properties of the $(30, 50; 100, 150, 40)$-fLMSS of Example 4.2.

$Y = \{u_{119}, u_{120}, \ldots, u_{168}\}$. The players in Y can reconstruct the secrets W_{40} and W_{80}, but no longer has sufficient shares to compute W_{120}.

Lemma 4.2 implies that any single player is associated with either $\lfloor \frac{150}{40} \rfloor$ or $\lceil \frac{150}{40} \rceil$ shares. In this example,

$$\frac{\ell}{d} = \frac{150}{40} = 3.75,$$

so each individual player is associated with either 3 or 4 shares. For example, consider the player u_{210}. This player must hold shares for the windows W_{80}, W_{120}, W_{160} and W_{200} and hence is an example of a player who holds four shares. On the other hand, player u_{235} must hold a secret for $W_{120}.W_{160}$ and W_{200} and thus holds three shares. Both players are illustrated in Figure 3.

Theorem 4.3 states that the total size of the shares of each player in a $(t_1, t_2; n, \ell, d)$-fLMSS is

$$\left\lceil \frac{\ell}{d} \right\rceil \frac{\log_2 |K|}{k - t}$$

bits. Here, we have a $(30, 50; 100, 150, 40)$-fLMSS, so if we are wishing to have a uniformly generated 32 bit secret (for example), we obtain

$$\left\lceil \frac{\ell}{d} \right\rceil \frac{\log_2(p)}{k - t} = \left\lceil \frac{150}{40} \right\rceil \frac{32}{50 - 30} = 4 \times \frac{32}{20} < 7,$$

so we would require each player to store at most 7 bits. For a secret of 64 bits, we would require at most 13 bits of storage. In comparison, were we to use Construction 3 each player would be required to store 100 shares each of size 32 bits, leading to a total storage of 3200 bits, or 6400 bits in the case of 64 bit secrets.

5 Application to key distribution for RFID-enabled supply chains

In this section we consider the application of fLMSS to the distribution of keys for RFID tags used in supply-chain management. Radio Frequency Identification (RFID) is a technology that uses radio signals to identify objects [1]. An RFID system consists of a tag, a reader and a back-end server [1]. The devices are used in transportation, logistics, manufacturing and processing. Typical applications include inventory control, animal tagging, postal tracking, airline baggage management, access control, and manufacturing processes [17].

RFID tags enable the identification, tracking and verification of products in a supply chain both automatically and in real time [1], hence their use is becoming more prevalent in manufacturing. They have the potential to store information such as batch numbers, date of manufacture, and so on. In supply chains, the predominant RFID standard is known as the Electronic Product Code (EPC) [6]. EPC tags can be regarded as new generation bar-codes that emit a code containing four elements [10]: a *header*, which denotes the EPC version number; a *domain manager* that details the manufacturer; an *object class* that specifies the item type; and a *serial number*, which is a universal identifier for the item. The unique serial number enables the tag to be linked to a database containing other vital information related to the product. Storing the EPC rather than all the information relevant to the product requires less memory on the tag, which is ideal as tags have a limited memory of up to 2KB of data [10].

However, the use of these tags creates new security and privacy challenges: for instance, a consumer who is in possession of a tagged product runs the risk that a passerby could scan the tag and thereby determine that they are carrying the product. There are many items such as medications for which this is potentially undesirable. One further complication is that there may not exist any prior secure channels (in the form of shared secret keys or similar) between the manufacturer and agents, such as merchants, who legitimately require access to the tags' contents. One way to address this is to exploit differences in the way in which a legitimate user is able to access the data on a tag or tags, as opposed to the potentially more restricted access available to a casual adversary. Juels, Pappu and Parno consider the following scenario:

Scenario 2 (Distributing a key over the RFID tags on all the items in a case [10])**.** *Suppose a manufacturer places tags on items that are shipped in bulk to a merchant who then sells them individually to customers. The merchant (who has a legitimate need to read the tag identities) has access to many tags. On the other hand, after the items are sold, an adversary (who should be prevented from recovering the tag identities) is expected to only have access to a small number of tags.*

In this scenario, the items are assumed to be shipped by the case-load to the merchant, and Juels *et al.* suggest the use of a secret sharing scheme to distribute a key across all the tags in a case of items, with each tag receiving a single share. This ensures that the merchant who possesses the entire case can gather enough shares to recover the identity, but once the items have been sold they are sufficiently dispersed that the identity can no longer be recovered. This is an example of a situation where a threshold (or ramp) secret sharing scheme gives a natural solution to the problem of securing identities of RFID tags from adversaries with access to a small number of tags, while allowing merchants who possess entire cases of tags to access the relevant information.

Juels *et al.* propose keeping the share sizes small by using a variation of a scheme due to Krawczyk that provides computational (rather than information-theoretic) security [11]. In Krawczyk's original scheme a large secret is encrypted with a (relatively) short key and the resulting ciphertext is shared using a $(0, k; n)$-ramp scheme, while the key is shared using a $(k; n)$-threshold scheme. Juels *et al.* suggest sharing both the key and the ciphertext with a

secret sharing scheme based on an error-correcting code, and claim that this enables them to *'make the size of our shares independent of the secret'*. In fact this is not correct, nor is their assertion that Krawczyk's scheme has *'shares with lengths independent of the secret's size'*. Rather, as Krawczyk states in his abstract [11], his scheme is 'an m-threshold scheme... in which shares corresponding to a secret S are of size $\frac{|S|}{m}$ plus a short piece of information whose length does not depend on the secret size but just in the security parameter. (The bound of $\frac{|S|}{m}$ is clearly optimal if the secret is to be recovered from m shares.)'.

The use of error correcting codes for secret sharing has been long studied [15, 13]; in particular it is known that a length n code with distance d and dual distance d^* gives a $(t_1, t_2; n-1)$-ramp scheme with $t_1 = d^* - 2$ and $t_2 = n - d + 1$ (see, for example, [16] for details). Thus if the dual distance of the code is small, then t_1 is also small and the resulting scheme only guarantees protection of the secret against small coalitions of players. In Krawczyk's scheme the computation security is ensured by the fact a threshold scheme is used to share the key. Replacing the threshold scheme in this construction by one based on an error correcting code leads to a similar reduction in security as would be caused by simply using a ramp scheme to share the message directly, does not reduce the storage relative to this more straightforward approach, and offers only computational rather than information-theoretic security guarantees. As such, this error-correcting code approach does not appear to offer any clear advantages in this context.

5.1 Instantiating a solution for Scenario 2

In order to use a secret-sharing based scheme for Scenario 2, it is necessary to preload the appropriate data on all the tags. Once a case has been ordered, the supplier can set up the appropriate tags and attach them to all the items in the case. However, this may be time consuming and inconvenient. Instead, it would be more convenient to be able to attach the tags to the products as they come off the production line prior to packing. This leads to the following scenario, which is essentially that considered by Juels *et al.* in their Example 2 [10].

Scenario 3 (Setting up a sequence of RFID tags for items coming off a production line)**.** *Suppose a manufacturer attaches RFID tags to items as they come off the production line. The items are then packed and shipped to meet orders coming in from wholesale customers. The manufacturer wishes to distribute keys across items in an order using a $(t; n)$-threshold secret sharing scheme, as in Scenario 2. However, the customers may order differing numbers of items, and at the time when the data is being placed on the tags the manufacturer does not yet know what these orders are going to be (either in terms of their sizes or to which customer they will be shipped). The shares on the tags in a single order must enable the wholesaler to recover a suitable key, yet adversaries who obtain fewer than t shares from a given order should learn no information about the key. (In particular, this means that the tags from a certain wholesaler's orders should not allow that wholesaler to learn the key corresponding to another wholesale customer's order.)*

Juels *et al.* propose the use of a *Sliding-Window-Information-Secret-Sharing (SWISS) scheme* for this purpose. Their basic SWISS scheme uses a $(k; n)$-threshold scheme with key window length $\ell = 2n$ and offset $d = n$. They observe that this can be generalised to $\ell = \frac{(\Psi+1)n}{\Psi}$ with $d = \frac{n}{\Psi}$ for $\Psi < n$. They further note that in place of a threshold scheme they could use the secret sharing scheme they developed for Scenario 2.

The fact that the choice of Ψ completely determines ℓ and d once n is known is unnecessarily restrictive here. We observe that the requirements of Scenario 3 are in fact essentially the same as those of Scenario 1; hence, a $(t_1, t_2; n, \ell, d)$-fLMSS is an appropriate solution for the manufacturer's needs in this situation.

Example 5.1. In Example 4.2, we saw that the use of a $(30, 50; 100, 150, 40)$-fLMSS allowed a 64 bit secret to be distributed while only requiring each player to store a 13 bit share. This is well within the capacity of an EPC Gen2 RFID tag, and would be suitable, for example, in a situation where merchants order shipments of at least 100 items at a time.

Advantages of using a fLMSS for this application include the following:

- The direct use of a ramp scheme rather than an arbitrary error-correcting code explicitly gives the values of the important parameters, so that the resulting trade-off between security and efficiency is entirely clear. Furthermore, the use of Construction 2 in the fLMSS gives the essential property that both the sharing and the secret recovery can be efficiently performed.
- The fLMSS provides information-theoretic security rather than relying on computational assumptions.
- The use of Construction 5 reduces the number of unauthorised sets who can access a given secret relative to a SWISS scheme of analogous parameters.
- By separating the window length ℓ from the offset d, we have enabled a more flexible choice of parameters that allows for the appropriate security/efficiency trade-off to be chosen to directly suit application requirements.

The systematic analysis of the various components of the fLMSS in Section 4 ensures that the trade-offs inherent in the selection of parameters are explicit and well-understood, making the fLMSS a widely applicable tool for applications of this nature.

6 Conclusion

Localised multisecret sharing is a natural concept with a range of potential applications. We have showed that a fLMSS scheme provides a flexible and lightweight tool for approximating the ideal behaviour of a $(t; n)$-LMSS in a restricted environment such as that provided by the use of RFID tags. Interesting open problems would be to find further applications for these schemes, and to determine whether their security can be further enhanced through improvements to Construction 5.

References

[1] S. Abughazalah, K. Markantonakis, and K. Mayes. Enhancing the key distribution model in the RFID-enabled supply chains. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pages 871–878. IEEE, 2014.

[2] G. R. Blakley. Safeguarding cryptographic keys. *Managing Requirements Knowledge, International Workshop on*, 0:313, 1979.

[3] C. Blundo, A. D. Santis, G. D. Crescenzo, A. G. Gaggia, and U. Vaccaro. Multi-secret sharing schemes. In Y. Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 150–163. Springer, 1994.

[4] C. Blundo, A. D. Santis, and U. Vaccaro. Efficient sharing of many secrets. In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, *STACS*, volume 665 of *Lecture Notes in Computer Science*, pages 692–703. Springer, 1993.

[5] R. M. Capocelli, A. D. Santis, L. Gargano, and U. Vaccaro. On the size of shares for secret sharing schemes. In J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 101–113. Springer, 1991.

[6] EPCglobal. Radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 mhz–960 mhz version 1.0. 9. http://www.epcglobalinc.org, 2004.

[7] J. Herranz, A. Ruiz, and G. Sáez. New results and applications for multi-secret sharing schemes. *Designs, Codes and Cryptography*, pages 1–24, 2013.

[8] M. Ito, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. In *IEEE Globecom*, pages 99–102, 1987.

[9] W.-A. Jackson, K. M. Martin, and C. M. O'Keefe. Multisecret threshold schemes. In D. R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 126–135. Springer, 1993.

[10] A. Juels, R. Pappu, and B. Parno. Unidirectional key distribution across time and space with applications to RFID security. In P. C. van Oorschot, editor, *USENIX Security Symposium*, pages 75–90. USENIX Association, 2008.

[11] H. Krawczyk. Secret sharing made short. In D. R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 136–146. Springer, 1993.

[12] K. Martin and W.-A. Jackson. A combinatorial interpretation of ramp schemes. *Australasian Journal of Combinatorics*, 14:51–60, 1996.

[13] J. Massey. Minimal codewords and secret sharing. In *Proceedings 6th Joint Swedish-Russian International Workshop on InformationTheory*, pages 276–279, 1993.

[14] B. Masucci. Sharing multiple secrets: Models, schemes and analysis. *Des. Codes Cryptography*, 39(1):89–111, 2006.

[15] R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Commun. ACM*, 24(9):583–584, 1981.

[16] M. B. Paterson and D. R. Stinson. A simple combinatorial treatment of constructions and threshold gaps of ramp schemes. *Cryptography and Communications*, 5(4):229–240, 2013.

[17] C. M. Roberts. Radio frequency identification (RFID). *Computers & Security*, 25(1):18–26, 2006.

[18] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.