Evolving Secret Sharing Schemes Based on Polynomial Evaluations and Algebraic Geometry Codes

Chaoping Xing and Chen Yuan

Shanghai Jiao Tong University, Shanghai, China

Abstract. A secret sharing scheme enables the dealer to share a secret among n parties. A classic secret sharing scheme takes the number n of parties and the secret as the input. If n is not known in advance, the classic secret sharing scheme may fail. Komargodski, Naor, and Yogev [7, TCC 2016] first proposed the evolving secret sharing scheme that only takes the secret as the input. In the work [7, TCC 2016], [8, TCC 2017] and [2, Eurocrypt 2020], evolving threshold and ramp secret sharing schemes were extensively investigated. However, all of their constructions except for the first construction in [2] are inspired by the scheme given in [7], namely, these schemes rely on the scheme for st-connectivity which allows to generate infinite number of shares.

In this work, we revisit evolving secret sharing schemes and present three constructions that take completely different approach. Most of the previous schemes mentioned above have more combinatorial flavor, while our schemes are more algebraic in nature. More precisely speaking, our evolving secret sharing schemes are obtained via either the Shamir secret sharing or arithmetic secret sharing from algebraic geometry codes alone. Our first scheme is an evolving k-threshold secret sharing scheme with share size $k^{1+\epsilon}\log t$ for any constant $\epsilon > 0$. Thus, our scheme achieves almost the same share size as in [7, TCC 2016]. Moreover, our scheme is obtained by a direct construction while the scheme in [7, TCC 2016] that achieves the $(k-1)\log t$ share size is obtained by a recursive construction which makes their structure complicated. Our second scheme is an evolving k_t-threshold secret sharing scheme with any sequence $\{k_t\}_{t=1}^{\infty}$ of threshold values that has share size t^4. This scheme improves the share size by $\log t$ given in [8] where a dynamic evolving k_t-threshold secret sharing scheme with the share size $O(t^4 \log t)$ was proposed. In addition, we also show that if the threshold values k_t grow in rate $\lfloor \beta t \rfloor$ for a real $\beta \in (0,1)$, then we have a dynamic evolving threshold secret sharing scheme with the share size $O(t^{4\beta})$. For $\beta < 0.25$, this scheme has sub-linear share size which was not known before. Our last scheme is an evolving $(\alpha t, \beta t)$-ramp secret sharing scheme with constant share size. One major feature of this ramp scheme is that it is multiplicative as the scheme is also an arithmetic secret sharing scheme. We note that the same technique in [8] can also transform all of our schemes to a robust scheme as our scheme is linear.[1]

[1] We note that by replacing the building block scheme with an arithmetic secret sharing scheme, the evolving $(\alpha t, \beta t)$-ramp secret sharing scheme in [2] can also be

1 Introduction

Secret sharing scheme enables the dealer to share a secret among n parties such that a authorized subset of parties can reconstruct the secret while a unauthorized subset of parties learns nothing about the secret. In this secret sharing scheme, to generate these n shares, the dealer must learn the number of parties in advance. If parties arrive at a different time, the dealer has to estimate the number of parties and generate all shares before the arrival of the first party. Since the share size, privacy and reconstruction all depend on the number n of parties, a pessimistic estimation of n may cause a large overhead in the share size or reconstruction and privacy while an underestimation of n will force the dealer to refresh the shares of all existing parties. Komargodski, Naor, and Yogev [7] proposed the evolving secret sharing scheme as a possible solution to this problem. In this scheme, the dealer does not need to know the number of parties in advance. When a new party arrives, the dealer could generate her share without updating the shares of all the existing parties. Since we handle an infinite number of parties, the share size tends to infinity as well for the threshold case. To measure the performance of an evolving secret sharing scheme, we write the share size as a function in the index of party, i.e., the t-th party has the share size $f(t)$. In [7], Komargodski, Naor, and Yogev observed that the share size of an evolving 2-threshold secret sharing scheme is at least $\log t + \log\log t$ which might be counterintuitive as the share size of the celebrated 2-threshold Shamir secret sharing scheme [9] is merely $\log t$. This feature distinguishes the evolving secret sharing scheme from the classic secret sharing scheme. Given the important role of secret sharing scheme as a cryptographic primitive and this new feature, we believe that evolving secret sharing scheme is worth a thorough investigation.

Before presenting our results, let us take a quick review at the known results. Like a classic secret sharing scheme, an evolving secret sharing scheme is k-threshold if any subset of k parties can reconstruct the secret and any subset of $k-1$ parties learn nothing about the secret. Komargodski, Naor, and Yogev [7] showed that an evolving k-threshold secret sharing scheme can have share size roughly equal to $(k-1)\log t$. Komargodski and Paskin-Cherniavsky [8] considered the evolving secret sharing scheme with dynamic threshold in which the threshold of the t-th party is k_t such that $k_1, k_2, \ldots$ is a non-decreasing sequence. They constructed an evolving secret sharing scheme with any sequence of threshold value in which the share size of the t-th party is $O(t^4 \log t)$. Beimel and Othman [1] extended this evolving threshold secret sharing scheme to evolving ramp secret sharing scheme. Let $b(t), g(t) : \mathbb{N} \to \mathbb{N}$ be two non-decreasing function. An evolving $(b(t), g(t))$-ramp secret sharing scheme is an evolving secret sharing scheme such that a subset of parties is authorized if it contains at least $g(t)$ parties from the first t parties for some $t \in \mathbb{N}$ and a subset of parties is unauthorized if it dose not contain more than $b(t)$ parties from the first t parties for any $t \in \mathbb{N}$. Beimel and Othman [1] considered the case $b(t)$ and $g(t)$ is a

multiplicative. However, their share size is much bigger than ours as each party hold multiple shares.

linear function. In this case, their evolving ramp secret sharing scheme has share size $O(1)$ which is almost as good as ramp secret sharing scheme. Beimel and Othman [2] further considered the case $g(t) - b(t) = O(t^\beta)$ for some $\beta \in (0, 1)$. Note that the evolving secret sharing scheme with dynamic threshold is also an evolving $(b(t), g(t))$-ramp secret sharing scheme if we set the sequence of threshold value to be $k_t = b(t) + 1$. However, they found that such small gap could lead to a significant improvement on the share size. The two schemes they proposed have the share size $O(t^{4-\frac{1}{\log^2 \beta}} \log^2 t)$ and $O(t^{\frac{1-\beta}{\beta}} \log t)$, respectively. We conclude this subsection by tabulating the known result below.

Reference	**Scheme**	**Share Size of the tth Party**
[7]	k-threshold	$O(k \log t)$
[8]	k_t-threshold	$O(t^4 \log t)$
[1]	$(\alpha t, \beta t)$-ramp for $0 < \alpha < \beta < 1$	$O(1)$
[2]	$(k, 2k)$-ramp	$O(\log k \log t)$
[2]	$(\gamma t - t^\beta, \gamma t)$-ramp for $0 < \gamma, \beta < 1$	$O(t^{4-\frac{1}{\log^2 \beta}} \log^2 t)$
[2]	$(\gamma t - t^\beta, \gamma t)$-ramp for $0 < \gamma, \beta < 1$	$O(t^{(1-\beta)/\beta} \log t)$

1.1 Our Results

We note that all known evolving threshold secret sharing scheme rely on the secret sharing scheme for st-connectivity which can be extended to any length. The scheme for st-connectivity is not multiplicative and thus might not be suitable for the applications like secure multi-party computation. In this work, we take a different approach which only relies on the multiplicative secret sharing scheme alone. Although our evolving threshold secret sharing scheme is also not multiplicative, we can show that this scheme is somewhat multiplicative under very restrictive conditions. Moreover, we believe that our evolving secret sharing scheme is the natural and straightforward extension of the classic secret sharing scheme. Our evolving ramp secret sharing scheme based on arithmetic secret sharing scheme is multiplicative.

We first propose two evolving k-threshold secret sharing schemes, one has simpler structure but larger share $k^2 \log t$ and another one is slightly complicated but has share size that can be arbitrarily close to $k \log t$. Then, we extend our technique to the evolving secret sharing scheme with dynamic threshold. Our scheme achieves the share size at most t^4 which is better than the previous construction [8] $O(t^4 \log t)$. In addition, for $k_t = \lfloor \beta t \rfloor$ with $\beta \in (0, 1)$ we obtain an evolving dynamic k_t-threshold secret sharing scheme with share size $O(t^{4\beta})$. We emphasize that in all of the above schemes, the secret of each Shamir secret sharing scheme is either the secret of this evolving secret sharing scheme or share of other Shamir secret sharing schemes. Our last contribution is an evolving $(\alpha t, \beta t)$-ramp secret sharing scheme with constants α, β and $O(1)$ share size. The approach we take is purely algebraic such that the share of each party is

actually an evaluation of an algebraic curve. Thus, the multiplicative property comes for free. However, the price we have to pay is a larger gap between α and β which usually does not affect the asymptotic performance of any secure multi-party computation protocol. In conclusion, we tabulate our results in the following table.

Reference	**Scheme**	**Share Size of the tth Party**	**Remark**
Theorem 2	k-threshold	$O(k^{1+\epsilon}\log t)$ for any real $\epsilon > 0$	almost same size as [7]
Theorem 3	k_t-threshold	$O(t^4)$	improvement to [8]
Theorem 4	$k_t = \lfloor \beta t \rfloor$-threshold	$O(t^{4\beta})$	new result
Theorem 6	$(\alpha t, \beta t)$-ramp for some $0 < \alpha < \beta < 1$	$O(1)$	same size as [1] but with multiplicativity

Note that all of our schemes are obtained via either the Shamir secret sharing scheme or the arithmetic secret sharing based on algebraic geometric codes. In [8], they proposed the robust version of the evolving secret sharing scheme and showed how to transform a linear evolving secret sharing scheme to its robust version. Since our new scheme keeps linearity, we can apply the same transformation to obtain a robust version of our evolving secret sharing scheme.

1.2 Our Techniques

In previous works, their approaches rely on the st-connectivity. In this work, we take a completely different approach. Our construction might be seen as a natural extension of the classic secret sharing scheme. Let us first look at our construction of the evolving k-threshold secret sharing scheme.

Constant Threshold. We partition parties into different generations like the other approaches. The party in the g-th generation holds k shares, $sh_1^g, \ldots, sh_k^g$ where sh_ℓ^g is a share generated by the ℓ-threshold Shamir secret sharing scheme. We create $k-1$ virtual parties for each generation. This virtual party is assigned the share in the same way as the real party except that they do not appear in the sequence of parties.[2] For $\ell < k$, the ℓ-threshold Shamir secret sharing scheme shares a secret which is a share held by the ℓ-th virtual party in the previous generation and the k-threshold Shamir secret sharing scheme shares the secret of this evolving secret sharing scheme. When it comes to the reconstruction, assume that an authorized set A has c_g parties from the g-th generation and g is the largest generation index in A. Then, these c_g shares can recover c_g shares held

[2] Their share will be secret shared among parties in the next generation. Then, the secret reconstructed by the parties in the next generation will not be collided with the shares held by the parties in this generation.

by c_g virtual parties from the $(g-1)$-th generation according to our scheme. We can replace these c_g parties with the new c_g virtual parties in our reconstruction. Since the share of virtual parties is equally effective as the share of real party, we move to an earlier generation without loss of any shares. We can continue in this manner until we collect k shares in the same generation. The similar argument also works for the privacy.

This scheme is simple but not very efficient. The secret of the Shamir secret sharing scheme is too big i.e., the field size of Shamir secret sharing scheme from the adjacent generation has to increase by k times. This yields an evolving secret sharing scheme with share size $k^2 \log t$. To reduce the share size, our improved scheme connects the g-th generation with the $(g-r)$-th generation, i.e., the secret shared among parties in the g-th generation is the share held by the virtual parties in the $(g-r)$-th generation. This modification can reduce the share size to $k^{1+\frac{1}{r}} \log t$. We still have a challenge to overcome. Note that such a scheme only works for the g-th generation with $g > r$. For the first r generations, we apply our original scheme since it will not affect our asymptotic performances. Then, there comes another challenge. For the i-th generation with $i \leq r$, their shares will be a secret for both of the $(i+1)$-th generations and $(i+r)$-th generations. To avoid that these secrets are overlapped, we create $2k-2$ virtual parties, $k-1$ of them are used for the $(i+1)$-th generation and another $k-1$ of them are used for the $(i+r)$-th generation. This will settle all the issues.

Dynamic Threshold. We move to our construction of evolving secret sharing scheme with dynamic threshold. We present our scheme in two steps. As usual, we partition parties into different generations.

- Our first scheme referred to as the basic scheme only handles the case that the threshold value in the same generation does not change, i.e., the threshold value for the g-th generation is k_g. Each party in the g-th generation hold k_{g-1} shares, $sh^g_1, \ldots, sh^g_{k_{g-1}}$ where sh^g_ℓ is a share generated by the $(\ell + k_g - k_{g-1})$-threshold Shamir secret sharing scheme Π_ℓ. For $1 \leq \ell \leq k_{g-1} - 1$, Π_ℓ shares a secret which is a share held by the ℓ-th virtual party in the previous generation and $\Pi_{k_{g-1}}$ shares the secret of this evolving secret sharing scheme. Therefore, there are $k_{g-1} - 1$ virtual parties in the $(g-1)$-th generation. Regarding the reconstruction, assume that there are c_g parties from the g-th generation. They can recover $c_g - (k_g - k_{g-1})$ shares held by the virtual parties from the $(g-1)$-th generation. Note that we lose $k_g - k_{g-1}$ shares during such recovery while the threshold value also changes from k_g to k_{g-1}. Note that the set of original parties is authorized, i.e., the number of parties in the first g-th generation is at least k_g. When we replace these c_g parties from g-th generation with $c_g - (k_g - k_{g-1})$ virtual parties from the $(g-1)$-th generation, this resulting set is still authorized. One can continue in this manner until all shares are from the same generations. Then, we are done. The privacy argument works almost in the same way.
- In our second step, we first set the threshold k_g to be the threshold value of the last party in the g-th generation and run the basic scheme to generate

the shares. Note that the virtual parties now have the same threshold value as the last party in their generation. Let $k_{g,t}$ be the threshold value of the t-th party in the g-th generation. For the t-th party in the g-th generation, we use $k_{g,t} - k_{g-1}$ Shamir secret sharing schemes $\Pi_{k_{g-1}}, \ldots, \Pi_{k_{g,t}}$, each will generate t shares that are assigned to the first t parties in this generation. For $k_{g,t} - k_{g-1} \leq \ell \leq k_{g,t}$, Π_ℓ is the ℓ-threshold Shamir secret sharing scheme with a secret that is a share held by the $(\ell - k_{g,t} + k_{g-1})$-th virtual party defined in the basic scheme. The reconstruction works almost in the same way as the basic scheme. Let the t-th party in the g-th generation be the last party in our authorized set A and assume that there are c_g parties from the g-th generation in A. Then, these c_g parties in the g-th generation can recover $c_g - (k_{g,t} - k_{g-1})$ shares held by virtual parties in the $(g-1)$-th generation. Replace them with these $c_g - (k_{g,t} - k_{g-1})$ virtual parties in A. One can check that this resulting set is still authorized. One can continue in this manner until all shares are from the same generations. Then, we are done. The privacy argument works similarly and we refer interested readers to Theorem 3 for details.

Ramp Case. The technique we employ for the evolving ramp secret sharing scheme is different. We still partition parties into different generations. For any party in the g-th generation, the share is an evaluation of an algebraic curve C_g, i.e., each party is assigned an evaluation of this algebraic curve at one point. Roughly speaking, C_g yields an arithmetic secret sharing scheme Π_g over $\mathbb{F}_q$ with $q = p^2$ on N_g parties such that Π_g has $(\gamma - \epsilon)N_g$-uniformity and $(\gamma + \epsilon)N_g$-reconstruction where $N_g \approx pN_{g-1}$ and γ is any constant in $(\frac{1}{p} + \epsilon, 1 - \epsilon)$.[3] The first g generations now consist of N_g parties. When the first party in the g-th generation arrives, we invoke Π_g for secret s to generate N_g shares such that its first N_{g-1} shares match the shares held by the first $(g-1)$ generations. [4] The parties in the g-th generation are assigned the remaining $N_g - N_{g-1}$ shares. We claim that such evolving ramp secret sharing scheme has $(\gamma - \frac{1}{p} - \epsilon)t$-privacy and $(\frac{p(\gamma+\epsilon)}{1+(p-1)(\gamma+\epsilon)})t$-reconstruction. The privacy argument is clear since our arithmetic secret sharing scheme Π_g has $(\gamma - \epsilon)N_g$-uniformity. After fixing the first N_{g-1} shares, this scheme still has $(\gamma - \frac{1}{p} - \epsilon)N_g \geq (\gamma - \frac{1}{p} - \epsilon)t$ privacy for any $t \leq N_g$. The reconstruction divides into two cases:

1. The party from the g-th generation with index $t \leq (1+(p-1)(\gamma+\epsilon))N_{g-1}$: by definition an authorized set including the t-th party must contain $(\frac{p(\gamma+\epsilon)}{1+(p-1)(\gamma+\epsilon)})t$ parties prior to her. That means, this authorized set contains at least $(\gamma + \epsilon)N_{g-1}$ parties from the first $g-1$ generations. Running the reconstruction scheme of Π_{g-1} can recover the secret.
2. Otherwise, we have $(\frac{p(\gamma+\epsilon)}{1+(p-1)(\gamma+\epsilon)})t \geq (\gamma+\epsilon)N_g$. Running the reconstruction scheme of Π_g can recover the secret.

[3] The $\epsilon = \frac{2}{\sqrt{q}-2}$ gap is caused by the genus of this curve.

[4] s is the secret to be shared.

The multiplicative claim comes from the observation that the arithmetic secret sharing scheme Π_g is multiplicative.

2 Preliminaries

In this section, we present the definition of evolving secret sharing scheme and some notations that will be used later. We use log to represent the logarithmic function with base 2. $\mathbb{F}_q$ is a finite field with q elements. $\mathbb{N} = \{1, 2, \ldots\}$ be the collections of all positive integers. We denote by $[n]$ the set $\{1, \ldots, n\}$. We use the notation 2^S to denote the collections of all subsets of S. $\mathcal{T} \subseteq 2^{[n]}$ is monotone if for any $A \in \mathcal{T}$ and $B \supseteq A$, $B \in \mathcal{T}$.

2.1 Access structure

To define the secret sharing scheme, we first need to define the access structure.

Definition 1 (Access Structures [2]) *An access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ over n parties $1, \ldots, n$ consists of a pair of collections of subsets in $2^{[n]}$ such that $\mathcal{T}_{Yes}$ and $2^{[n]} \setminus \mathcal{T}_{No}$ are monotone and $\mathcal{T}_{Yes} \cap \mathcal{T}_{No} = \emptyset$. The set in $\mathcal{T}_{Yes}$ is called an authorized set and the set in $\mathcal{T}_{No}$ is called an unauthorized set.*

A secret sharing scheme realizes an access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ if the unauthorized set of parties will learn nothing about the secret while the authorized set of parties can reconstruct the secret. The formal definition is given as follows.

Definition 2 (Secret Sharing Scheme) *A secret sharing scheme for an access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ consists of a pair of algorithms (SHARE, RECON). SHARE is a probabilistic algorithm that takes the secret $s \in S$ as an input and generate n shares $(sh_1^s, sh_2^s, \ldots, sh_n^s)$, i.e., $SHERE(s) = (sh_1^s, \ldots, sh_n^s)$. RECON is a deterministic algorithm that takes the input of shares of a subset $B \subseteq [n]$ and output a string in S. The requirements are:*

1. *CORRECTNESS: The secret can be reconstructed by any authorized set, i.e., for any secret $s \in S$, any share with $SHERE(s, r) = (sh_1^s, \ldots, sh_n^s)$ and any $A \in \mathcal{T}_{Yes}$, we have $RECON(\{sh_i^s\}_{i \in A}, A) = s$.*
2. *PRIVACY: For any two secrets $s_1 \neq s_2$ and any unauthorized set $B \in \mathcal{T}_{No}$, the distributions of shares $\{sh_i^{s_1}\}_{i \in B}$ and $\{sh_i^{s_2}\}_{i \in B}$ are identical. The probability is over the randomness of the SHARE.*

The share size of this scheme is the maximum number of bits each party holds in the worst case over all parties and all secrets.

Definition 3 (Threshold Access Structures) *An access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ over n parties $1, \ldots, n$ is called an k-threshold access structure if $\mathcal{T}_{Yes} = \{A \subseteq [n] : |A| \geq k\}$ and $\mathcal{T}_{No} = \{B \subseteq [n] : |B| \leq k - 1\}$.*

It is well known that the Shamir secret sharing scheme can realize any threshold access structure.

Claim 1 (Shamir [9]) *Given a secret $s \in \mathbb{F}_q$, the Shamir secret sharing scheme can realize k-threshold access structure over n parties for any $n \leq q$. More precisely, this secret sharing scheme generates n shares $(sh_1, \ldots, sh_n) \in \mathbb{F}_q \times \mathbb{F}_q \cdots \times \mathbb{F}_q$. We call such a scheme a k-threshold Shamir secret sharing scheme.*

Definition 4 (Ramp Access Structures) *An access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ over n parties $1, \ldots, n$ is called an (b, g)-ramp access structure if $\mathcal{T}_{Yes} = \{A \subseteq [n] : |A| \geq g\}$ and $\mathcal{T}_{No} = \{B \subseteq [n] : |B| \leq b\}$.*

A (b, g)-ramp secret sharing scheme is a secret sharing scheme realizing the (b, g)-ramp access structure.

2.2 Evolving Access Structure

Next, we proceed to the definition of evolving access structure introduced in [7].

Definition 5 (Evolving Access Structures [2]) *An Evolving access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ is a pair of collections of sets $\mathcal{T}_{Yes}, \mathcal{T}_{No} \subseteq 2^{\mathbb{N}}$ such that every set in $\mathcal{T}_{Yes} \cup \mathcal{T}_{No}$ is finite. Moreover, for every $t \in \mathbb{N}$, the intersection $\mathcal{T}_t := (\mathcal{T}_{Yes} \cap 2^{[t]}, \mathcal{T}_{No} \cap 2^{[t]})$ is an access structure defined in Definition 1.*

We can also define the evolving secret sharing scheme accordingly since an evolving secret sharing scheme is used to realize an evolving access structure.

Definition 6 (Evolving Secret Sharing Scheme) *Let $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ be an evolving access structure and $\mathcal{T}_t$ be the intersection in Definition 5. Let S be the secret domain. A secret sharing scheme Π for S and $\mathcal{T}$ consists of two algorithms (SHARE, RECON). They satisfy the following requirements.*

1. *SHARE is a probabilistic algorithm generating shares one by one, i.e., it takes the secret $s \in S$ and the shares of first $t-1$ parties as inputs and outputs the t-th share,*

$$SHARE(s, sh_1^s, sh_2^s, \ldots, sh_{t-1}^s) = sh_t^s$$

   where $sh_1^s, sh_2^s, \ldots, sh_{t-1}^s$ are the first $t-1$ shares.
2. *CORRECTNESS: For every t, every secret $s \in S$ and every subset $A \in \mathcal{T}_{Yes} \cap 2^{[t]}$, we have $RECON(\{sh_i^s\}_{i \in A}, A) = s$.*
3. *SECURITY: For every t, any two secrets $s_1 \neq s_2 \in S$ and every subset $B \in \mathcal{T}_{No} \cap 2^{[t]}$, the distributions of shares $\{sh_i^{s_1}\}_{i \in B}$ and $\{sh_i^{s_2}\}_{i \in B}$ are identical. The probability is over the randomness of the SHARE.*

Similarly, we can define the evolving threshold access structure and evolving ramp access structure.

Definition 7 (Evolving Threshold Access Structures) *An evolving access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ is called an evolving k-threshold access structure if for every $t \in \mathbb{N}$, $\mathcal{T}_{Yes} \cap 2^{[t]} = \{A \in 2^{[t]} : |A| \geq k\}$ and $\mathcal{T}_{No} \cap 2^{[t]} = \{B \in 2^{[t]} : |B| \leq k-1\}$.*

Definition 8 (Evolving Ramp Access Structures) *An evolving access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ is an evolving (b, g)-ramp access structure if for every $t \in \mathbb{N}$, $\mathcal{T}_{Yes} \cap 2^{[t]} = \{A \in 2^{[t]} : |A| \geq g\}$ and $\mathcal{T}_{No} \cap 2^{[t]} = \{B \in 2^{[t]} : |B| \leq b\}$.*

A secret sharing scheme realizes evolving k-threshold ((b, g)-ramp) access structure is called evolving k-threshold ((b, g)-ramp resp) secret sharing scheme.

In [8], they can also consider an evolving access structure $\mathcal{T}$ with dynamic threshold, i.e., the threshold grows with the time. Note that such definition is a generalization of evolving access structure and is thus even more difficult to be realized. Recall that $\mathcal{T}_t = (\mathcal{T}_{Yes} \cap 2^{[t]}, \mathcal{T}_{No} \cap 2^{[t]})$.

Definition 9 (Evolving Access Structures with dynamic threshold) *An evolving access structure $\mathcal{T}$ has a sequence of thresholds $k_1, k_2, \ldots$ if for every $t \in \mathbb{N}$ $\mathcal{T}_t$ is a k_t-threshold access structure over t parties.*

Remark 1 *There is an equivalent definition for this access structure. Let $\mathcal{T}$ be an evolving access structure with a sequence of thresholds $k_1, k_2, \ldots$. Given a subset of parties $A = \{i_1, \ldots, i_r\}$, then*

- *A is authorized if there exists an $\ell \leq r$ with $\ell \geq k_{i_\ell}$.*
- *A is unauthorized if for any $\ell \leq r$, it holds $\ell < k_{i_\ell}$.*

In [1], they propose a ramp version of this access structure. They consider the evolving ramp access structure with its gap growing with the time.

Definition 10 (Evolving ramp Access Structures with dynamic gap [1]) *Let $b(t)$ and $g(t)$ be the two non-decreasing functions. An evolving access structure $\mathcal{T} = (\mathcal{T}_{Yes}, \mathcal{T}_{No})$ is an evolving $(b(t), g(t))$-ramp access structure if for every $t \in \mathbb{N}$, $\mathcal{T}_{Yes} \cap 2^{[t]} = \{A \in 2^{[t]} : |A| \geq g(t)\}$ and $\mathcal{T}_{No} \cap 2^{[t]} = \{B \in 2^{[t]} : |B| \leq b(t)\}$.*

The evolving secret sharing schemes realizing these access structures are named accordingly.

We can also define the evolving arithmetic secret sharing scheme by considering the square of an evolving secret sharing scheme. In this work, we only consider the evolving arithmetic ramp secret sharing scheme. The formal definition is given as follows.

Definition 11 *Π is an evolving arithmetic $(b(t), g(t), h(t))$-ramp secret sharing scheme if the followings hold,*

1. *Π realizes an evolving $(b(t), g(t))$-ramp access structure.*

2. *For every $t \in \mathbb{N}$ and any subsets $A \subseteq [t]$ of size $h(t)$, we have*

$$\mathrm{RECON}(\{sh_i^a sh_i^b\}_{i\in A}, A) = ab,$$

where $(sh_1^a, \ldots, sh_t^a)$ and $(sh_1^b, \ldots, sh_t^b)$ are the shares corresponding to secret a and b respectively. In other words, one can recover the product of two secrets by given the component-wise product of any $h(t)$ shares.

3 Evolving Threshold Secret Sharing Scheme

In this section, we present the evolving secret sharing schemes realizing the evolving threshold access structure and the evolving access structure with dynamic threshold.

3.1 Constant Threshold Case

We first present a weak version of our evolving k-threshold secret sharing scheme with share size at most $k^2 \log t$. This weak version will be a basic scheme for our final version which achieves share size $k^{\frac{r+1}{r}} \log t$ for any integer r. Although this scheme is slightly worse than the state-of-the-art scheme proposed in [7], there is a prominent feature in our scheme which does not hold in any other constructions. Our construction only makes use of Shamir secret sharing scheme which preserves the multiplicative property. Such property is a key ingredient for secure multi-party computation. We observe that in very restrictive condition, our scheme is somewhat multiplicative. Besides, the approach we take for the evolving threshold secret sharing scheme also sheds a light on the construction of evolving secret sharing scheme with dynamic threshold. This observation leads to a most efficient evolving secret sharing scheme with dynamic threshold. Our share size is $\log t$ multiplicative factor smaller than that of the scheme in [8].

Theorem 1. *There exists an evolving k-threshold secret sharing scheme of share size $k^2 \log t$. Moreover, the share of each party consists of k shares generated by the Shamir secret sharing scheme.*

Proof. Like the approach in other works, we partition parties into different generations. We use the term the j-th party in the i-th generation to specify this party. One can easily convert it to its index in the sequence of all parties. The set of parties in the i-th generation is $\{k, k+1, \ldots, \ldots, q^{k^{i+1}}\}$. The first $k-1$ slots are left for the "virtual" parties on purpose. Those shares held by the virtual parties in this generation will be used as a secret to be shared among parties in the next generation.

Let s be the secret to be shared. We start with 0-th generation. The dealer does the following:

– Share the secret s via a k-threshold Shamir secret sharing scheme over $\mathbb{F}_{q^k}$ and denote its shares by $(sh_1^0, sh_2^0, \ldots, sh_{q^k}^0)$.

– Assign the share sh_t^0 to the t-th party in the 0-th generation.

Note that $sh_1^0, \ldots, sh_{k-1}^0$ are not held by any party from the 0-th generation. These $k-1$ shares will be used as a secret to be shared among the parties in the next generation.

Now, we turn to the description of the g-th generation. Since the share of parties in the $(g-1)$-th generation has been generated, denote by $\mathbf{sh}_t^{g-1}$ the share held by the t-th party in the $(g-1)$-th generation. When the first party in the g-th generation arrives, the dealer does the following

– For $j < k$, share the secret $\mathbf{sh}_j^{g-1}$ via a j-threshold Shamir secret sharing scheme over $\mathbb{F}_{q^{k^{g+1}}}$ and denote by its shares $(sh_{j,1}^g, sh_{j,2}^g, \ldots, sh_{j,q^{k^{g+1}}}^g)$.
– Share the secret s via a k-threshold Shamir secret sharing scheme and denote its shares by $(sh_{k,1}^g, sh_{k,2}^g, \ldots, sh_{k,q^{k^{g+1}}}^g)$.
– Assign the share $\mathbf{sh}_t^g = (sh_{1,t}^g, sh_{2,t}^g, \ldots, sh_{k,t}^g)$ to the t-th party in the g-th generation.

In general, the t-th party (either virtual or real) in the g-th generation holds the share $\mathbf{sh}_t^g = (sh_{1,t}^g, sh_{2,t}^g, \ldots, sh_{k,t}^g)$ for $t = 1, \ldots, q^{k^{g+1}}$.

We proceed to the reconstruction. Assume that we have an authorized set of size k in which there are c_i parties from i-th generation. Assume g is the biggest index with $c_g > 0$. If $c_g = k$, we are done as the secret is shared among parties in the g-th generation via a k-threshold Shamir secret sharing scheme. Otherwise, according to our scheme, the c_g parties from the g-th generation can recover c_g secrets $\mathbf{sh}_1^{g-1}, \ldots, \mathbf{sh}_{c_g}^{g-1}$ which are shares held by c_g virtual parties from the $(g-1)$-th generation. During this process, we obtain c_g new shares for the $(g-1)$-th generation by sacrificing the c_g shares for the g-th generation. We can continue in this manner until that we collect k shares belonging to the same generation. Then, we can recover the secret by running the reconstruction algorithm of k-threshold Shamir secret sharing scheme.

The privacy argument is rather simple. We observe that since in each generation, the secret s is only shared via a k-threshold Shamir secret sharing scheme. Since any $k-1$ parties can only recover at most $k-1$ shares for the same generation, the parties in the unauthorized set will learn nothing about the secret.

As for the share size, we observe that for the g-th generation, the secret size for each Shamir secret sharing scheme is at most $k \times k^g \log q = k^{g+1} \log q$ which is at most the share size. Thus, the Shamir secret sharing scheme over $\mathbb{F}_{q^{k^{g+1}}}$ is big enough to share such secret. Since we only use k Shamir secret sharing schemes in the g-th generation, the share size is $k^{g+2} \log q$ for each party in this generation. The first party in the g-th generation is at least the q^{k^g}-th party in the sequence of all parties. Thus, the share size at most $k^2 \log t$ for the t-th party in the sequence of all parties.

Remark 2 Although this scheme only invokes the Shamir secret sharing scheme, it is not multiplicative. However, we can modify this construction to make it

somewhat multiplicative. There is a tool called reverse multiplication friendly embedding [3] which can map an element in the extension field to a vector in the base field while keeping the multiplication. A pair (ϕ, ψ) is called an $(k, m)_q$-reverse multiplication friendly embedding if $\phi : \mathbb{F}_q^k \to \mathbb{F}_{q^m}$ and $\psi : \mathbb{F}_{q^m} \to \mathbb{F}_q^k$ are two $\mathbb{F}_q$-linear maps satisfying

$$\mathbf{x} * \mathbf{y} = \psi(\phi(\mathbf{x}) \cdot \phi(\mathbf{y}))$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^k$ and $*$ is a component-wise product. Note that the secret of our Shamir secret sharing scheme is $\mathbf{sh}_t^g = (sh_{1,t}^g, sh_{2,t}^g, \ldots, sh_{k,t}^g)$. ϕ can map $\mathbf{sh}_t^g$ to an element in the extension field while keeps the multiplication. It remains to show that we can collect enough shares so as to recover the product of the secrets. Assume that A is a set containing $2k - t$ parties in the g-th generation and $2t$ parties in the $(g + 1)$-th generation. Note that each party in our scheme holds k Shamir secret sharing scheme whose threshold value is $1, \ldots, k$. Thus, the product of shares held by these $2t$ parties can recover the product of the secrets that is shared by the ℓ-threshold Shamir secret sharing scheme for $\ell = 1, \ldots, t$. These t product of the secrets together with $2k - t$ parties in the g-th generation can recover the product of the secrets of this evolving secret sharing scheme. By slight modification to our scheme, we can show that it is possible to obtain the same result for parties in two non-adjacent generations. As a comparison, the previous threshold constructions can only recover the product of the secrets if there are at least $2k$ parties in the same generation. Since the application of this scheme is very restrictive, we will not expand it in detail.

This scheme is simple but not very efficient. We next show how to modify our scheme to reduce the share size to any number close to $k \log t$. Observe that the share size of the i-th generation is k times bigger than the share size of the $(i-1)$-th generation. This happens because we use the share held by the virtual parties from the $(i - 1)$-th generation as a secret shared among parties in the i-th generation. To reduce the share size, we need to connect the parties in the i-th generation with the virtual parties in earlier generation.. More precisely, we first apply this evolving k-threshold secret sharing scheme described above to generate the shares of our first $r + 1$ generations. Then, for the party starting from the g-th generation with $g > r$, the secret they share is the share held by the virtual parties from the $(g - r)$-th generation instead of the $(g - 1)$-th generation. We will see that this modification can significantly bring down the growth rate of our field size.

Theorem 2. *For any integer $r > 0$, there exists an evolving k-threshold secret sharing scheme of share size $k^{\frac{r+1}{r}} \log t$. Moreover, the share of each party consists of k shares generated by the Shamir secret sharing scheme.*

Proof. The sharing algorithm for the first $r + 1$ generations is exactly the same as that in our previous theorem except that we leave $2k - 2$ slots for the virtual parties. We briefly review this scheme. The set of parties in the i-th generation is $\{2k-1, 2k, \ldots, \ldots, q^{k^{i+1}}\}$. The first $2k-2$ slots are left for the virtual parties

on purpose. When the first parties in the i-th generation with $i \leq r$ arrives, the dealer does the following,

- For $j < k$, share the secret $\mathbf{sh}_{k-1+j}^{i-1}$ via a j-threshold Shamir secret sharing scheme over $\mathbb{F}_{q^{k^{i+1}}}$ and denote its shares by $(sh_{j,1}^i, sh_{j,2}^i, \ldots, sh_{j,q^{k^{i+1}}}^i)$.
- Share the secret s via a k-threshold Shamir secret sharing scheme sharing over $\mathbb{F}_{q^{k^{i+1}}}$ and denote its shares by $(sh_{k,1}^i, sh_{k,2}^i, \ldots, sh_{k,q^{k^{i+1}}}^i)$.
- Assign $\mathbf{sh}_t^i = (sh_{1,t}^i, \ldots, sh_{k,t}^i)$ to the t-th party in the i-th generation.

Note that $\mathbf{sh}_t^i = (sh_{1,t}^i, \ldots, sh_{k,t}^i)$ for $i = 1, \ldots, 2k-2$ are held by $2k-2$ virtual parties. The first $k-1$ shares will become secrets to be shared among parties in the $(i+r)$-th generation and the second $k-1$ shares will become secrets to be shared among parties in the $(i+1)$-th generation.

Next we proceed to the g-th generation with $g > r$. Let $\ell_g = \lfloor k^{r+1+\frac{g-r}{r}} \rfloor$. It holds that $\frac{\ell_g}{\ell_{g-1}} = k^{\frac{1}{r}}$. The set of parties in the g-th generation is $\{k, k+1, \ldots, q^{\ell_g}\}$ while the set of virtual parties is {1,. . . ,k-1}. When the first party in the g-th generation arrives, the dealer does the following

- For $j < k$, share a secret $\mathbf{sh}_j^{g-r}$ via a j-threshold Shamir secret sharing scheme over $\mathbb{F}_{q^{\ell_g}}$ and denote its shares by $(sh_{j,1}^g, sh_{j,2}^g, \ldots, sh_{j,q^{\ell_g}}^g)$.
- Share the secret s via a k-threshold Shamir secret sharing scheme over $\mathbb{F}_{q^{\ell_g}}$ denote its shares by $(sh_{k,1}^g, sh_{k,2}^g, \ldots, sh_{k,q^{\ell_g}}^g)$.
- Assign $\mathbf{sh}_t^g := (sh_{1,t}^g, \ldots, sh_{k,t}^g)$ to the t-th party in the g-th generation.

We proceed to the proof of reconstruction. Assume that we have an authorized set A of size k in which there are c_g parties from the g-th generation such that g is the largest integer with $c_g > 0$. If $c_g \geq k$, we can recover the secret immediately. Otherwise, if $g > r$, these c_g shares can recover c_g shares $\mathbf{sh}_1^{g-r}, \ldots, \mathbf{sh}_{g-r}^{c_g}$ held by c_g virtual parties in the $(g-r)$-th generation. This means we obtain c_g new shares for the $(g-r)$ generation by sacrificing the c_g shares for the g-th generation. Then, we move to the $(g-1)$-th generation. One can continue in this manner until either collecting k shares for the same generation or reaching the r-th generation. For the former case, we can recover the secret immediately. For the latter case, we obtain shares held by the parties from the first $r+1$ generations. Note that some of them might be held by the virtual parties. However, there are in total still k shares held by k different parties as this set is authorized. Observe that for $g < r$, the $k-1$ secrets shared among parties in the $(g+r)$-th generation $\mathbf{sh}_1^g, \ldots, \mathbf{sh}_{k-1}^g$ is different from the secret shared among parties in the $(g+1)$-th generation $\mathbf{sh}_k^g, \ldots, \mathbf{sh}_{2k-2}^g$. This ensures that the secrets we recover from different generations will not collide with each other. We can now safely apply the same reconstruction algorithm of the evolving secret sharing scheme in our previous theorem to recover the secret s.

The privacy argument is obvious since when we recover the shares we lose the same amount of shares. For example, let A be an unauthorized set in which there are c_i parties from the i-th generation. Let g be the biggest index with

$c_g > 0$. Since A is not authorized, we have $c_g < k$. All they can do is to use these c_g shares to recover c_g secrets which are the shares of parties from earlier generation. Then, we can discard these c_g shares from the g-th generation. Now, we still got the same amount of shares but we move to an earlier generation. Thus, we can apply the induction to reach the conclusion.

As for the share size, we first show that the secret size is at most the share size for any of our threshold Shamir secret sharing scheme. For the g-th generation with $g \leq r$, we can simply apply the same argument in our previous theorem. For the g-th generation with $g > r$, as the secret is the share for the $(g-r)$-th generation, the secret size for each Shamir secret sharing scheme is at most $k \times \ell_{g-r} \log q \leq \ell_g \log q$. Thus, the Shamir secret sharing scheme over $\mathbb{F}_{q^{\ell_g}}$ is big enough to share such secret. Finally, we turn to bound the share size. For the first party in the g-th generation with $g > r$, her index in the sequence of all parties is at least $q^{\ell_{g-1}}$ while her share size is $\ell_g \log q$. Thus, the share size of our scheme is at most $k^{\frac{r+1}{r}} \log t$.

3.2 Dynamic Threshold Case

Theorem 3. *For any sequence of threshold value $\{k_1, k_2, \ldots, k_t, \ldots\}$ that define a dynamic access structure, there exists an evolving secret sharing scheme for sharing one bit secret in which the share size of the t-th party is at most t^4.* [5]

Proof. As usual, we partition the parties into different generation. First of all, we assume that the threshold value for parties in each generation is the same. This assumption will make our scheme easy to describe. Then, we will show how to extend it to handle different threshold values in the same generation. Let $a_i = 2^{2^{i+1}}$ and we have $a_{i+1} = a_i^2$.

The simplified case: In this case, we assume that the threshold value in the same generation is the same, i.e., let k'_i be the threshold value of the i-th generation. As usual, we use the term the j-th party in the i-th generation to specify this party. One can easily convert it to its index in the sequence of all parties. The set of parties in the i-th generation is $\{1, \ldots, a_i\}$. The set of virtual parties in the i-th generation is $\{a_i + 1, \ldots, a_i + k'_i - 1\}$. Denote by $\mathbf{sh}_t^i$ the share held by the t-th party in the i-th generation. When the first party in the i-th generation arrives, the dealer does the following

- For $\ell \in \{1, \ldots, k'_{i-1} - 1\}$, share a secret $\mathbf{sh}_\ell^{i-1}$ via an $(\ell + k'_i - k'_{i-1})$-threshold Shamir secret sharing scheme over $\mathbb{F}_{2^{a_{i-1}}}$. This Shamir secret sharing scheme yields $a_i + k'_i - 1$ shares $(sh^i_{\ell,1}, \ldots, sh^i_{\ell,a_i}, \ldots, sh^i_{\ell,a_i+k'_i-1})$.
- Share the secret s via a k'_i-threshold Shamir secret sharing scheme. This Shamir secret sharing scheme yields $a_i + k'_i - 1$ shares $(sh^i_{k'_{i-1},1}, \ldots, sh^i_{k'_{i-1},a_i+k'_i-1})$.

[5] We emphasize that there is no log factor in our bound. This is because our field size is exponentially bigger than the number of parties which incurs an t multiplicative factor.

– Assign the share $\mathbf{sh}_t^i := (sh_{1,t}^i, \ldots, sh_{k'_{i-1},t}^i)$ to the t-th party in the i-th generation.

For 0-th generation, the dealer only invoke a k'_0-threshold Shamir secret sharing scheme to share secret s since there does not exist any generation ahead of it. The shares $\mathbf{sh}_t^i$ for $t = a_i + 1, \ldots, a_i + k'_i - 1$ are held by the t-th virtual parties in the i-th generation.

We now move to the reconstruction argument. Assume that A is an authorized set that contains c_i parties from the i-th generation. Let f be the smallest index with $c_f > 0$. Since A is authorized, there must exist generation index g such that $\sum_{i=f}^{g} c_i \geq k'_g$ and for any $\ell < g$, we have $\sum_{i=f}^{\ell} c_i < k'_\ell$. First of all, from these two inequalities, we obtain that $c_g > k'_g - k'_{g-1}$. This implies that these c_g parties from the g-th generation can recover $c_g - (k'_g - k'_{g-1})$ shares $\mathbf{sh}_{a_{i-1}+1}^{i-1}, \ldots, \mathbf{sh}_{a_{i-1}+c_g-(k'_g-k'_{g-1})}^{i-1}$. Now, we have $\sum_{i=f}^{g} c_i - (k'_g - k'_{g-1}) \geq k'_{g-1}$ shares for the first $(g-1)$ generations. It follows that A is still an authorized set. We can continue in this manner until we arrive at the f-th generation. Then, we obtain $\sum_{i=f}^{g} c_i - (k'_g - k'_f) \geq k'_f$ shares held by the parties in the f-th generation. It is clear that we can now reconstruct the secret s with these shares.

Regarding the privacy, assume that A is an unauthorized set that contains c_i parties from the i-th generation. Let g be the largest index with $c_g > 0$. Then, for any $\ell \leq g$, $\sum_{i=0}^{\ell} c_i < k'_\ell$. As we know, to reconstruct the secret s, we need at least k'_ℓ shares from the parties in the ℓ-th generation. It is clear that the c_g parties in g-th generation alone can not reconstruct the secret s as $c_g < k'_g$. In this case, these c_g parties in g-th generation can recover at most $c_g - (k'_g - k'_{g-1})$ shares held by virtual parties in the $(g-1)$-th generation. We now have $(\sum_{i=0}^{g-1} c_i) + c_g - (k'_g - k'_{g-1}) < k'_{g-1}$ shares for the first $g-1$ generations. We can apply the induction to finish our proof.

Regarding the share size, we first note that the field size of our Shamir secret sharing scheme is big enough. The secret shared among the g-th generation has the form

$$\mathbf{sh}_\ell^{g-1} = (sh_{1,\ell}^{g-1}, \ldots, sh_{k'_{g-2},\ell}^{g-1}) \in \mathbb{F}_{2^{a_{g-2}}}^{k'_{g-2}}.$$

Thus, the size of our secret is at most $k'_{g-2} \times \log(2^{a_{g-2}}) < a_{g-2} \times a_{g-2} = a_{g-1}$. Moreover, since the party in the g-th generation hold $k'_{g-1} \leq a_{g-1}$ shares of Shamir secret sharing scheme, the share held by each party in the g-th generation is of size at most $a_{g-1} \log 2^{a_{g-1}} = a_{g-1}^2$. Thus, the share size of the first party in this generation is at most t^2.

The general case: We now proceed to the general case that the threshold value for parties in the same generation may vary. We refer the (g,t)-th party to the t-th party in the g-th generation. It is clear that (i_1, t_1)-th party arrives before (i_2, t_2) if $i_1 < i_2$ or $i_1 = i_2$ and $t_1 < t_2$. One can see that it is a totally ordered set. The set of parties in the g-th generation is $\{1, 2, \ldots, a_g\}$. We denote by $k_{(g,t)}$ as the threshold value of (g,t)-th party. We set k'_g be the threshold

value for the last party in the g-th generation, i.e., $k'_g = k_{(g,a_g)}$. The set of virtual parties in the g-th generation is $\{a_g + 1, \ldots, a_g + k'_g - 1\}$ which means they are placed at the end of this generation. By setting the threshold value of virtual parties k'_g, we can treat these virtual parties as the last real party in this generation.

When the first party in the g-th generation arrives, the dealer does the following:

- Invoke the evolving secret sharing scheme described in the simplified case to generate the share $\mathbf{sh}^g_r := (sh^g_{1,r}, \ldots, sh^g_{k'_{g-1},r})$ for $r \in \{1, \ldots, a_g + k'_g - 1\}$. Assign $\mathbf{sh}^g_r$ to the r-th party in this generation.
- Let t range over $[1, a_g - 1]$ and do the following:
  - For $\ell \in \{1, \ldots, k'_{g-1}\}$, share the secret $\mathbf{sh}^{g-1}_{a_{g-1}+\ell}$(the share held by the ℓ-th virtual party in the $(g-1)$-th generation) via an $(\ell + k_{(g,t)} - k'_{g-1})$-threshold Shamir secret sharing scheme over $\mathbb{F}_{2^{a_{g-1}}}$. This Shamir secret sharing scheme yields t shares $(s^t_{\ell,1}, \ldots, s^t_{\ell,t})$.
  - Assign the (g,t)-th party the share of the form

$$\left(\left(s^t_{1,t}, \ldots, s^t_{k'_{g-1},t}\right), \ldots, \left(s^{a_g-1}_{1,t}, \ldots, s^{a_{g-1}}_{k'_{g-1},t}\right), \mathbf{sh}^g_t\right)$$

    .

For party in the 0-th generation, the dealer only keeps the virtual party that shares the secret s. We note that the share held by the last party and all the virtual parties in each generation is exactly the same as they hold in the simplified case.

The reconstruction works as follows. Assume that there is an authorized set $A = \{(i_1, t_1), (i_2, t_2), \ldots, \}$ such that there are c_i parties from the i-th generation in A and $(i_1, t_1) < (i_2, t_2) < \cdots$ is ordered. Since A is authorized, there must exist an r such that $r \geq k_{(i_r,t_r)}$ and $\ell < k_{(i_\ell,t_\ell)}, \forall \ell < r$. Let $b = |\{(i_r, t) \in A : t \leq t_r\}|$, i.e., the number of parties from the t_r-th generation in A that arrives no later than (i_r, t_r)-th party. Observe that $r = b + \sum_{i=i_1}^{i_r-1} c_i \geq k_{(i_r,t_r)}$ and $\sum_{t=i_1}^{i_r-1} c_t < k'_{i_r-1}$. This implies $b > k_{(i_r,t_r)} - k'_{i_r-1}$ and thus these b parties from the i_r-th generation can recover $b - (k_{(i_r,t_r)} - k'_{i_r-1})$ shares held by the virtual parties in the (i_r-1)-th generations. We replace these b parties in A with $b-(k_{(i_r,t_r)}-k'_{i_r-1})$ virtual parties in the (i_r-1)-th generation. The resulting set A is of size $r-(k_{(i_r,t_r)}-k'_{i_r-1}) \geq k'_{i_r-1}$. Now, we can invoke the reconstruction scheme from the simplified case to recover the secret as A is still an authorized set.

Regarding the privacy issue, assume $A = \{(i_1, t_1), \ldots, (i_r, t_r)\}$ is any unauthorized set with c_i parties from the i-th generation. This means that $\ell < k_{i_\ell,t_\ell}$ for any $\ell \leq r$. We first argue that (i_r, t_r)-th party does not help to recover the secret. It is clear that shares held by the c_{i_r} parties from the i_r-th generation alone can not reconstruct secret s. The only thing the (i_r, t_r)-th party can do is to work with another $c_{i_r} - 1$ parties from the same generation to recover at most

$c_{i_r} - (k_{(i_r,t_r)} - k'_{i_r-1})$ shares held by the virtual parties in the (i_r-1)-th generation. Now, we can replace these c_{i_r} parties with these $c_{i_r} - (k_{(i_r,t_r)} - k'_{i_r-1})$ virtual parties as their shares are already used. We treat these $c_{i_r} - (k_{(i_r,t_r)} - k'_{i_r-1})$ virtual parties the same as the last parties in the (i_r-1)-th generation since they have the same threshold value. Compared with our previous set A, the only difference is this replacement. We note that the privacy condition still holds as

$$c_{i_r} - (k_{(i_r,t_r)} - k'_{i_r-1}) + \sum_{t=0}^{i_r-1} c_t = r - (k_{(i_r,t_r)} - k'_{i_r-1}) < k'_{i_r-1}.$$

The induction now works because we reduce the generation index of its last party in A. Therefore, we can claim that the (i_r, t_r)-th party does not help to recover the secret. Remove this party from set A and we can apply the same argument to the (i_{r-1}, t_{r-1})-th party. This desired result follows from the induction.

Regarding the share size, we first note that the field size of our Shamir secret sharing scheme is big enough. The same argument from the simplified case show that the size of our secret is at most $k'_{g-2} \times \log 2^{a_{g-2}} < a_{g-2} \times a_{g-2} = a_{g-1}$.[6] Moreover, since the party from g-th generation hold at most $k'_{g-1} \times a_g \leq a_{g-1}a_g$ shares of Shamir secret sharing scheme over $\mathbb{F}_{2^{a_{g-1}}}$. As the index of the first party in the g-th generation is at least a_{g-1}, the share size of this party is at most t^4.

We can also consider the condition that the sequence of threshold value is a function in t, the index of the party. If this function grows slowly, we can reduce the share size.

Theorem 4. *Let $\{k_1, k_2, \ldots, k_t, \ldots\}$ that define a dynamic access structure. If $k_t = \lfloor t^\beta \rfloor$ for some constant $\beta \in (0,1)$, there exists an evolving secret sharing scheme for sharing one bit secret in which the share size of t-th party is at most $O(t^{4\beta})$.*

Proof. We use the scheme in the general case in Theorem 3 with some modifications. Recall that $a_g = 2^{2^{g+1}} = a_{g-1}^2$. We have that the threshold value of the (g,t)-th party is $k_{(g,t)} \leq a_g^\beta$.[7] The Shamir secret sharing scheme used in the g-th generation is now defined over $\mathbb{F}_{2^{\lfloor a_{g-1}^\beta \rfloor}}$ instead of $\mathbb{F}_{2^{a_{g-1}}}$ since the secret it share is $\mathbf{sh}_r^{g-1} := (sh_{1,r}^{g-1}, \ldots, sh_{k'_{g-2},r}^{g-1}) \in \mathbb{F}_{2^{\lfloor a_{g-2}^\beta \rfloor}}^{k'_{g-2}}$ whose share size is at most $k'_{g-2} \times \lfloor a_{g-2}^\beta \rfloor < \lfloor a_{g-1}^\beta \rfloor$. We observe that if two adjacent parties (g,r)-th party and $(g,r+1)$-th party have the same threshold value, they can share the same Shamir secret sharing scheme. This is exactly what we do in the simplified case. In this sense, the sharing algorithm simply skip the iteration $t = r$ and go straight to $t = r+1$. Observe that there are at most t^β different threshold values

[6] The threshold value must be smaller than the index of this party.

[7] For simplicity, we assume that the last party in the g-th generation is the a_g-th party in the sequence. This will not change the asymptotic property of our share size

among t parties. This reduce the number of shares from $k'_g \times a_g$ to $k'_g \times k'_g = a_g^{2\beta}$. Now, the share size of the first party in the g-th generation is at most $O(t^{4\beta})$.

Remark 3 We note that all of above scheme can be transformed to a robust evolving secret sharing scheme by applying the transformation in [8] as our scheme is linear. Let us sketch the proof. We only apply the Shamir secret sharing scheme which is a linear secret sharing scheme. Clearly, it holds for the first generation. By induction, we can show that for the g-th generation, this Shamir secret sharing scheme preserves the linear relationship between the secret and the share. This completes the proof. The same argument can also be applied to the ramp evolving secret sharing scheme presented in the next section.

4 Evolving Ramp Secret Sharing Schemes

If we want to construct ramp secret sharing schemes with constant-sized shares, it is natural to consider secret sharing schemes based on function fields, or e-quivalently algebraic geometry codes as Chen and Cramer did in [4].

Let us first introduce function fields of one variable over finite fields and algebraic geometry codes very briefly. The reader may refer to the books [10, 11] for the details on this topic. For convenience of the reader, we start with some background on global function fields over finite fields. The reader may refer to [10] for detailed background on function fields and algebraic-geometric codes.

For a prime power q, let $\mathbb{F}_q$ be the finite field of q elements. An algebraic function field over $\mathbb{F}_q$ in one variable is a field extension $F \supset \mathbb{F}_q$ such that F is a finite algebraic extension of $\mathbb{F}_q(x)$ for some $x \in F$ that is transcendental over $\mathbb{F}_q$. The field $\mathbb{F}_q$ is called the full constant field of F if the algebraic closure of $\mathbb{F}_q$ in F is $\mathbb{F}_q$ itself. Such a function field is also called a global function field. From now on, we always denote by $F/\mathbb{F}_q$ a function field F with the full constant field $\mathbb{F}_q$.

A discrete valuation of $F/\mathbb{F}_q$ is a map from F to $\mathbb{Z} \cup \{+\infty\}$ satisfying certain properties (see [10, Definition 1.19]). Then each discrete valuation ν from $F/\mathbb{F}_q$ to $\mathbb{Z} \cup \{+\infty\}$ defines a valuation ring $O = \{f \in F : \ \nu(f) \geq 0\}$ that is a local ring [10, Theorem 1.1.13]. The maximal ideal P of O is given by $P = \{f \in F : \ \nu(f) > 0\}$ and it is called a *place*. We denote the valuation ν and the local ring O corresponding to P by ν_P and O_P, respectively. The residue class field O_P/P, denoted by F_P, is a finite extension of $\mathbb{F}_q$. The extension degree $[F_P : \mathbb{F}_q]$ is called *degree* of P, denoted by $\deg(P)$. A place of degree one is called a *rational* place. For a nonzero function $z \in F$, the principal divisor of z is defined to be $\mathrm{div}(z) = \sum_{P \in \mathbb{P}_F} \nu_P(z)P$. The zero and pole divisors of z are defined to be $\mathrm{div}(z)_0 = \sum_{\nu_P(z)>0} \nu_P(z)P$ and $\mathrm{div}(z)_\infty = -\sum_{\nu_P(z)<0} \nu_P(z)P$, respectively. Then we have $\deg(\mathrm{div}(z)) = 0$, i.e, $\deg(\mathrm{div}(z)_0) = \deg(\mathrm{div}(z)_\infty)$. For two functions $f, g \in F$ and a place P, we have $\nu_P(f+g) \geq \min\{\nu_P(f), \nu_P(g)\}$ and the equality holds if $\nu_p(f) \neq \nu_P(g)$ (note that $\nu_P(0) = +\infty$). This implies that $f + g \neq 0$ if $\nu_P(f) \neq \nu_P(g)$.

If F is the rational function field $\mathbb{F}_q(x)$, then every discrete valuation of $F/\mathbb{F}_q$ is given by either ν_∞ or $\nu_{p(x)}$ for an irreducible polynomial $p(x)$, where ν_∞ is defined by $\nu_\infty(f/g) = \deg(g) - \deg(f)$ and $\nu_{p(x)}(f/g) = a - b$ with $p(x)^a || f$ and $p(x)^b || g$ for two nonzero polynomials $f, g \in \mathbb{F}_q[x]$. It is straightforward to verify that the degrees of places corresponding to ν_∞ and $\nu_{p(x)}$ are 1 and $\deg(p(x))$, respectively.

Let $\mathbb{P}_F$ denote the set of places of F. The divisor group, denoted by $\mathrm{Div}(F)$, is the free abelian group generated by all places in $\mathbb{P}_F$. An element $D = \sum_{P \in \mathbb{P}_F} n_P P$ of $\mathrm{Div}(F)$ is called a divisor of F, where $n_P = 0$ for almost all $P \in \mathbb{P}_F$. We denote n_p by $\nu_P(D)$. The support, denoted by Supp(D), of D is the set $\{P \in \mathbb{P}_F : \ n_P \neq 0\}$. Thus, Supp(D) of a divisor D is always a finite subset of $\mathbb{P}_F$. For a divisor D of $F/\mathbb{F}_q$, we define the Riemann-Roch space associated with D by

$$\mathcal{L}(D) := \{f \in F^* : \ \mathrm{div}(f) + D \geq 0\} \cup \{0\},$$

where F^* denotes the set of nonzero elements of F. Then $\mathcal{L}(D)$ is a finite dimensional space over $\mathbb{F}_q$ and its dimension $\dim_{\mathbb{F}_q} \mathcal{L}(D)$ is determined by the Riemann-Roch theorem which gives

$$\dim_{\mathbb{F}_q} \mathcal{L}(D) = \deg(D) + 1 - \mathfrak{g} + \dim_{\mathbb{F}_q} \mathcal{L}(W - D),$$

where $\mathfrak{g}$ is the genus of F and W is a canonical divisor of degree $2\mathfrak{g}-2$. Therefore, we always have that $\dim_{\mathbb{F}_q} \mathcal{L}(D) \geq \deg(D) + 1 - \mathfrak{g}$ and the equality holds if $\deg(D) \geq 2\mathfrak{g} - 1$ [10, Theorems 1.5.15 and 1.5.17].

As we have to regenerate the secret and shares of the previous generation in a coming generation, we need the following lemma.

Lemma 2. *Let $F/\mathbb{F}_q$ be a function field of genus $\mathfrak{g}$. Let $Q, P_1, P_2, \ldots, P_t$ be $t+1$ pairwise distinct rational places. If $m \geq t + 2\mathfrak{g} - 1$, then the set*

$$\{(f(P_1), f(P_2), \ldots, f(P_t)) : \ f \in \mathcal{L}(mQ)\}$$

is equal to $\mathbb{F}_q^t$.

Proof. Consider the map

$$\pi : \ \mathcal{L}(mQ) \to \mathbb{F}_q^t; \quad f \mapsto (f(P_1), f(P_2), \ldots, f(P_t)).$$

Then π is $\mathbb{F}_q$-linear with kernel $\mathcal{L}\left(mQ - \sum_{i=1}^t P_i\right)$. Hence, by the Riemann-Roch Theorem, we have

$$\dim_{\mathbb{F}_q} \mathrm{Im}(\pi) = \dim_{\mathbb{F}_q} \mathcal{L}(mQ) - \dim_{\mathbb{F}_q} \mathcal{L}\left(mQ - \sum_{i=1}^t P_i\right) = m - \mathfrak{g} + 1 - (m - t - \mathfrak{g} + 1) = t.$$

This forces that $\mathrm{Im}(\pi) = \mathbb{F}_q^t$. The desired result follows.

Let $p \geq 5$ be a prime power and let $q = p^2$. Consider the Garicia-Stictenoth function field tower $\{F_i/\mathbb{F}_q\}$ given in Appendix A.

(1) For $i \geq 1$, put $\mathfrak{g}_i = \mathfrak{g}(F_i)$ and let $N_i = (q-1)p^{i-1}$. Label $N_i + 2$ $\mathbb{F}_q$-rational place of F_i by $Q_i, P_0^{(i)} P_1^{(i)}, P_2^{(i)}, \ldots, P_{N_i}^{(i)}$. Then we have $N_i/\mathfrak{g}_i \geq p-1$ for all $i \geq 2$.
(2) Fix an integer m_1 such that $\frac{m_1}{q-1} \in (\frac{1}{p} + \frac{2}{p-1}, 1)$. Put $\gamma = \frac{m_1}{q-1}$ and set $m_i = \gamma N_i$ for all $i \geq 1$.

Based on choice of the above parameters, for any $i \geq 1$ we have

$$\begin{aligned} m_{i+1} = \gamma N_{i+1} = N_i - N_i + \gamma N_{i+1} = N_i - \frac{1}{p} N_{i+1} + \gamma N_{i+1} \\ \geq N_i + \left(\gamma - \frac{1}{p}\right)(p-1)\mathfrak{g}_{i+1} \geq (1 + N_i) + 2\mathfrak{g}_{i+1} - 1. \end{aligned} \tag{1}$$

Now we construct our first evolving secret sharing scheme. Let $s \in \mathbb{F}_q$ be a secret.

- Generation 1: Randomly choose a function f_1 in the Riemann-Roch space $\mathcal{L}(m_1 Q_1)$ subject to $f_1(P_0^{(1)}) = s$. The shares are $\mathrm{sh}_i^{(1)} := f_1(P_i^{(1)})$ for $1 \leq i \leq N_1$. Note that this is just a Shamir secret sharing scheme.
- Generation 2: As $m_2 \geq N_1 + 2\mathfrak{g}_2$, by Lemma 2, we can randomly choose a function f_2 in the Riemann-Roch space $\mathcal{L}(m_2 Q_2)$ subject to $f_2(P_i^{(2)}) = f_1(P_i^{(1)})$ for $i = 0, 1, 2, \ldots, N_1$. The shares in generation 2 are $\mathrm{sh}_i^{(2)} := f_2(P_i^{(2)})$ for $1 \leq i \leq N_2$.
- Generation n: Continue in the above fashion and assume that we have constructed shares $\mathrm{sh}_i^{(n-1)}$ for $i = 1, 2, \ldots, N_{n-1}$. As $m_n \geq N_{n-1} + 2\mathfrak{g}_n$, by Lemma 2, we can randomly choose a function f_n in the Riemann-Roch space $\mathcal{L}(m_n Q_n)$ subject to $f_n(P_i^{(n)}) = f_{n-1}(P_i^{(n-1)})$ for $i = 0, 1, 2, \ldots, N_{n-1}$. The shares are $\mathrm{sh}_i^{(n)} := f_j(P_i^{(n)})$ for $1 \leq i \leq N_n$.

It is clear that the share size is $\log q = O(1)$. Furthermore, we claim:

the above scheme is a $(\alpha t, \beta t)$-ramp evolving secret sharing scheme for all $t \in N$, where $\alpha = \gamma - \frac{1}{p} - \frac{2}{p-1}$ and $\beta = \frac{p\gamma}{1+(p-1)\gamma}$.

Let us prove the above claim now.

(i) **Reconstruction:**
- First of all, we claim that the secret s can be reconstructed by any m_n+1 shares in $[N_n]$. This is because that $(f_n(Q_n), f_n(P_1^{(n)}, \ldots, f_n(P_{N_n}^{(n)})$ is a codeword of a q-ary $[N_n + 1, m_n - \mathfrak{g}_n + 1, \geq N_n + 1 - m_n]$-linear code.
- For $m_1 + 1 \leq t \leq N_1$, it is an $(m_1, m_1 + 1)$-threshold secret sharing scheme. Thus, $m_1 + 1 = \gamma N_1 \leq \beta t$ for all $\frac{\gamma}{\beta} N_1 \leq t \leq N_1$.
- Suppose that we have shown that the above secret sharing scheme has reconstruction βt from the shares in $[t]$ for all $t \leq N_{n-1}$. Now let reconstruction from shares in $[t]$ with $N_{n-1} + 1 \leq t \leq N_n$ for $n \geq 2$.
  Case 1. For $N_{n-1} + 1 \leq t \leq \frac{1-\gamma}{1-\beta} N_{n-1}$, let A be a subset of $[t]$ of size at least βt, then $A \cap [N_{n-1}]$ has size at least $|A| + N_{n-1} - t \geq$

$\beta t + N_{n-1} - t \geq \gamma N_{n-1} \geq m_{n-1} + 1$. Thus, s can be recovered by the shares $\{\mathrm{sh}_i^{(n)}\}_{i \in A \cap [N_n]} = \{\mathrm{sh}_i^{(n-1)}\}_{i \in A \cap [N_{n-1}]}$.
Case 2. For $\frac{1-\gamma}{1-\beta} N_{n-1} < t \leq N_n$, let A be a subset of $[t]$ of size at least βt. As $\beta = \frac{2\gamma}{1+\gamma}$, we have

$$\beta t > \beta \times \frac{1-\gamma}{1-\beta} N_{n-1} = p\gamma N_{n-1} = \gamma N_n \geq m_n.$$

Since any $m_n + 1$ shares can reconstruct the secret, the secret can be reconstructed by the shares in A. This completes the proof for reconstruction.

(ii) **Privacy:**

- For $m_1 + 1 \leq t \leq N_1$, it is an $(m_1, m_1 + 1)$-threshold secret sharing scheme. Thus, $m_1 \geq \gamma N_1 - 1 \geq \alpha t$ for all $t \leq N_1$.
- Let $n \geq 2$. For $N_{n-1} + 1 \leq t \leq N_n$, let A be a subset of $[t]$ of size at most αt, then

$$\deg \left(m_n Q_n - \sum_{i \in [N_{n-1}] \cup A} P_i^{(n)} - P_0^{(n)} \right) \geq m_n - (N_{n-1} + |A|) - 1.$$

As $m_n - (N_{n-1} + |A|) - 1 \geq \gamma N_n - N_{n-1} - \alpha t - 1 \geq 2\mathfrak{g}_n - 1$. Thus, there exists a function f in the set

$$\mathcal{L}\left(m_n Q_n - \sum_{i \in [N_{n-1}] \cup A} P_i^{(n)} \right) \setminus \mathcal{L}\left(m_n Q_n - \sum_{i \in [N_{n-1}] \cup A} P_i^{(n)} - P_0^{(n)} \right).$$

Hence, $f(P_0^{(n)}) \neq 0$ and $f(P_i^{(n)}) = 0$ for all $i \in [N_{n-1}] \cup A$. By multiplying a nonzero constant, we may assume that $f(P_0^{(n)}) = 1$. Now for any $s' \in \mathbb{F}_q$. Consider the function $f_n + (s' - s)f$. Then we have $(f_n + (s' - s)f)(P_0^{(n)}) = s'$, $(f_n + (s' - s)f)(P_i^{(n)}) = f_n(P_i^{(n)})$ for any $i \in [N_{n-1}] \cup A$. This shows privacy.

In conclusion, we have the following result.

Theorem 5. *Let $p \geq 5$ be a prime power and let $q = p^2$. Let γ be a real in the interval $(\frac{1}{p} + \frac{1}{p-1}, 1)$. Then there exists an $(\alpha t, \beta t)$-evolving ramp secret sharing scheme with secret and share sizes $\log q = O(1)$, where $\alpha = \gamma - \frac{1}{p} - \frac{1}{p-1}$ and $\beta = \frac{p\gamma}{1+(p-1)\gamma}$. Furthermore, the scheme (including share distribution and secret reconstruction) for any t players can be constructed in time $O(t^3)$.*

Theorem 6. *If $\gamma < \frac{1}{2}$, then the evolving ramp secret sharing scheme given in Theorem 5 is an evolving arithmetic $(\alpha t, \beta t, \delta t)$-ramp secret sharing scheme with $\delta = \frac{2p\gamma}{1-2\gamma+2p\gamma}$.*

Proof. Let A be a subset of $[t]$ with $|A| \geq \delta t$.

First, let us assume that $t \leq N_1$. Assume that shares in A are $\{f(P_i^{(1)})\}_{i\in A}$ and $\{g(P_i^{(1)})\}_{i\in A}$ for some $f, g \in \mathcal{L}(G_1)$. Then $\delta t > 2m_1$ for $t > \frac{2m_1}{\delta} = \frac{2\gamma}{\delta}N_1 = \frac{1-2\gamma+2p\gamma}{p}N_1$.

Now let us assume that $N_{n-1} < t \leq N_n$ for some $n \geq 2$. Assume that shares in A are $\{f(P_i^{(n)})\}_{i\in A}$ and $\{g(P_i^{(n)})\}_{i\in A}$ for some $f, g \in \mathcal{L}(G_n)$. We also assume that shares in $A \cap [N_1]$ are $\{f_1(P_i^{(n-1)})\}_{i\in A}$ and $\{g_1(P_i^{(n-1)})\}_{i\in A}$ for some $f_1, g_1 \in \mathcal{L}(G_{n-1})$.

Case 1. If $N_{n-1} + 1 \leq t < \frac{1-2\gamma}{1-\delta}N_{n-1}$, then $A \cap [N_{n-1}]$ has size at least $|A| + N_{n-1} - t \geq \delta t + N_{n-1} - t > 2\gamma N_{n-1} \geq 2m_{n-1} + 1$. Thus, the product $f_1(P_0^{(n-1)})g_1(P_0^{(n-1)})$ of the secrets can be recovered by the shares $\{f(P_i^{(n)})g(P_i^{(n)})\}_{i\in A\cap[N_n]} = \{f_1(P_i^{(n-1)})g_1(P_i^{(n-1)})\}_{i\in A\cap[N_{n-1}]}$.

Case 2. For $\frac{1-2\gamma}{1-\delta}N_{n-1} \leq t \leq N_n$, let A be a subset of $[t]$ of size at least βt. As $\delta = \frac{2p\gamma}{1-2\gamma+2p\gamma}$, we have

$$\delta t \geq \delta \times \frac{1-2\gamma}{1-\delta}N_{n-1} = 2p\gamma N_{n-1} = 2\gamma N_n \geq 2m_n.$$

Since any $2m_n + 1$ shares can reconstruct the product secret, the product secret can be reconstructed by the product shares in A. This completes the proof for reconstruction.

References

1. Amos Beimel and Hussien Othman. Evolving ramp secret-sharing schemes. In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 313–332. Springer, 2018.
2. Amos Beimel and Hussien Othman. Evolving ramp secret sharing with a small gap. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 529–555. Springer, 2020.
3. Ignacio Cascudo, Ronald Cramer, Chaoping Xing, and Chen Yuan. Amortized complexity of information-theoretically secure MPC revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 395–426. Springer, 2018.
4. Hao Chen and Ronald Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, pages 521–536, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

5. Arnaldo Garcia and Henning Stichtenoth. A tower of artin - schreier extensions of function fields attaining the drinfeld - vlâdut bound. *Inventiones Mathematicae*, 121:211–222, 01 1995.
6. Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of Number Theory*, 61:248C273, 12 1996.
7. Ilan Komargodski, Moni Naor, and Eylon Yogev. How to share a secret, infinitely. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 485–514, 2016.
8. Ilan Komargodski and Anat Paskin-Cherniavsky. Evolving secret sharing: Dynamic thresholds and robustness. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 379–393. Springer, 2017.
9. Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
10. Henning Stichtenoth. *Algebraic Function Fields and Codes*. Springer Publishing Company, Incorporated, 2nd edition, 2008.
11. M. A. Tsfasman and S. G. Vlăduţ. *Algebraic-Geometric Codes*. Springer, 1991.

A The Garcia-Stichtenoth tower

There are two function field towers by Garcia and Stichenoth [5, 6]. Let us make use of the tower given in [5].

Let p be a prime power and let $q = p^2$. The Garcia-Stichenoth tower given [5] is defined recursively as follows. Let $F_1 = \mathbb{F}_q(x_1)$ and for $n \geq 1$ let $F_{n+1} = F_n(z_{n+1})$, where z_2 satisfies the equation $z_2^p + z_2 = z_1^{q+1}$ and z_{n+1} satisfies the equation

$$z_{n+1}^p + z_{n+1} = \left(\frac{z_n}{x_{n-1}}\right)^{p+1} \tag{2}$$

for all $n \geq 2$.

Then the genus $\mathfrak{g}(F_n)$ of F_n is given by

$$\mathfrak{g}(F_n) = \begin{cases} p^n + p^{n-1} - p^{(n+1)/2} - 2p^{(n-1)/2} + 1 & \text{if } n \text{ is odd;} \\ p^n + p^{n-1} - \frac{1}{2}p^{n/2+1} - \frac{3}{2}p^{n/2} - p^{n/2-1} + 1 & \text{if } n \text{ is even.} \end{cases}$$

Furthermore, the number of $\mathbb{F}_q$-rational places of F_n is at least $(q-1)p^{n-1} + 2p$.