Lattice-Based Threshold-Changeability for Standard Shamir Secret-Sharing Schemes

Ron Steinfeld (Macquarie University, Australia)

(email: rons@ics.mq.edu.au)

Joint work with:

Huaxiong Wang (Macquarie University)

(email: hwang@ics.mq.edu.au)

Josef Pieprzyk (Macquarie University)

(email: josef@ics.mq.edu.au)

Overview

- (t,n)-Threshold Secret Sharing Schemes
  - Classical Shamir Scheme
- Changeable-Threshold Secret-Sharing Schemes
  - Drawbacks of previous solutions
- Our Approach: Lattice-Based Threshold-Changeability for Classical Shamir Scheme
  - Brief Review of Point Lattices
  - Method for increasing the threshold from t to $t' > t$
  - Lattice-based Decoding Algorithm & Correctness Analysis
  - Lattice-based Information-Theoretic Security Analysis

(t,n)-Threshold Secret Sharing

- Fundamental cryptographic scheme (Shamir, 1979)
  - Informal Definition:
    - A Dealer owning a secret *s* wishes to "distribute" knowledge of *s* among a group of *n* shareholders such that two conditions hold:
      - Correctness: Any subset of *t* shareholders can together recover *s*
      - Security: Any subset of less than *t* shareholders cannot recover *s*
- Many applications in information security – especially for achieving robustness of distributed security systems:
    - Consider an access control system with *n* servers
    - System is called t-robust if security is maintained even against attackers who succeed in breaking into up to t-1 servers
    - Can be achieved by distributing the access control secret among the n servers using a (t,n)-threshold secret sharing scheme.

(t,n)-Threshold Secret-Sharing

Definition 1 (Threshold Scheme) *A (t, n)-threshold secret-sharing scheme* $\mathsf{TSS} = (\mathsf{GC}, \mathsf{D}, \mathsf{C})$ *consists of three efficient algorithms:*

1 GC *(Public Parameter Generation): Takes as input a security parameter $k \in \mathcal{N}$ and returns a string $x \in \mathcal{X}$ of public parameters.*

2 D *(Dealer Setup): Takes as input $(k, x) \in \mathcal{N} \times \mathcal{X}$ and a secret $s \in \mathcal{S}(k, x) \subseteq \{0, 1\}^{k+1}$ and returns n shares $\mathbf{s} = (s_1, \ldots, s_n)$, where $s_i \in \mathcal{S}_i(k, x)$ for $i = 1, \ldots, n$. We denote by*

$$\mathsf{D}_{k,x}(., .) : \mathcal{S}(k, x) \times \mathcal{R}(k, x) \to \mathcal{S}_1(k, x) \times \cdots \times \mathcal{S}_n(k, x)$$

the mapping induced by algorithm D (here $\mathcal{R}(k, x)$ denotes the space of random inputs to D).

3 C *(Share Combiner): Takes as input $(k, x) \in \mathcal{N} \times \mathcal{X}$ and any subset $\mathbf{s}_I = (s_i : i \in I)$ of t shares, and returns a recovered secret $s \in \mathcal{S}(k, x)$. (here $I \subseteq [n]$ is a subset of size $\#I = t$).*

(t,n)-Threshold Secret-Sharing

- Classical Shamir Scheme (Shamir '79)

1. GC(k) (Public Parameter Generation):

   (a) Pick a (not necessarily random) prime $p \in [2^k, 2^{k+1}]$ with $p > n$.

   (b) Pick uniformly at random n distinct non-zero elements $\alpha = (\alpha_1, \ldots, \alpha_n) \in D((\mathbf{Z}_p^*)^n)$. Return $x = (p, \alpha)$.

2. D$_{k,x}(s, \mathbf{a})$ (Dealer Setup): To share secret $s \in \mathbf{Z}_p$ using $t-1$ uniformly random elements $\mathbf{a} = (a_1, \ldots, a_{t-1}) \in \mathbf{Z}_p^{t-1}$, build the polynomial

   $$a_{s,\mathbf{a}}(x) = s + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \in \mathbf{Z}_p[x; t-1].$$

   The ith share is $s_i = a(\alpha_i) \bmod p$ for $i = 1, \ldots, n$.

3. C$_{k,x}(\mathbf{s}_I)$ (Share Combiner): To combine shares $\mathbf{s}_I = (s_i : i \in I)$ for some $I \subseteq [n]$ with $\#I = t$, compute by Lagrange interpolation the unique polynomial $b \in \mathbf{Z}_p[x; t-1]$ such that $b(\alpha_i) \equiv s_i \pmod{p}$ for all $i \in I$. The recovered secret is $s = b(0) \bmod p$.

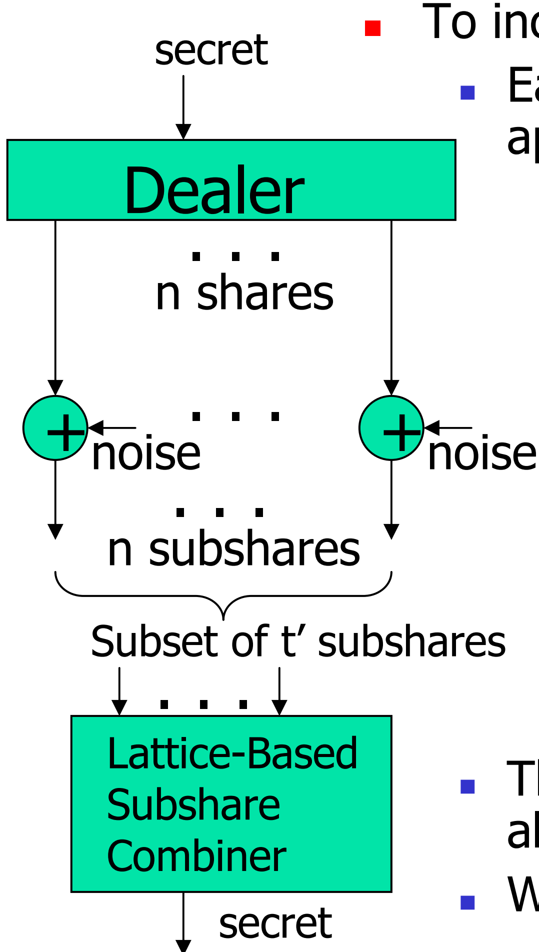
Changeable-Threshold Secret-Sharing

- Motivation:
  - In applications, choice of the threshold parameter t is a compromise between two conflicting factors:
    - Value of Protected System & Attacker Resources
      - → Pushing the threshold as high as possible
    - User Convenience and Cost
      - → Pushing the threshold as low as possible
  - Hence actual value of t will be an "equilibrium" value, which will change in time as the relative strength of the above conflicting factors change in time
- This motivates study of <u>Changeable-Threshold Secret-Sharing</u> schemes

Changeable-Threshold Secret-Sharing

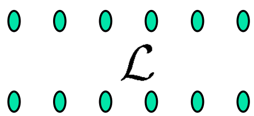
- Drawbacks of previous solutions are at least one of:
  - Dealer Involvement after setup phase [eg. Blundo'93]
    - Dealer broadcasts a message to all shareholders to allow them to update their shares from a (t,n) to a (t',n) scheme
    - <u>Implication:</u> Dealer must communicate after setup!
  - Initial (t,n)-threshold scheme is non-standard [eg. Martin'99]
    - Simple example: Dealer gives each shareholder two shares of the secret, one for a (t,n) scheme, another for a (t',n) scheme
    - <u>Implication:</u> Dealer must plan ahead!
  - Shareholders privately communicate with each other [eg. Desmedt'97]
    - E.g. Shareholders re-destribute secret among themselves for a (t',n) scheme via secure computation protocol
    - <u>Implication:</u> Shareholders must communicate!
- Our scheme does not have any of these drawbacks!
    - Although we only achieve relaxed correctness/security

Changeable-Threshold Secret-Sharing

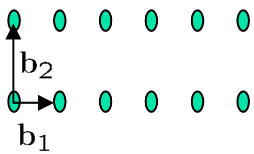


- Basic idea of our approach
  - To increase threshold from t to t′ > t,
    - Each Shareholder adds a random `noise′ integer (of appropriate size) to his share, to obtain a subshare
      - Subshares contain only partial information on original shares
      - → We expect that:
        - Any t subshares are not sufficient to recover secret
        - But t′ subshares (for some t′ > t depending on size of noise added) are sufficient to recover secret if we have an appropriate `error-correction algorithm′
        - (e.g if noise bit-length = ½ of share length, we expect that t′ ~ 2t subshares uniquely determine the secret)
    - The new `subshare combiner′ algorithm is the error correction algorithm
    - We construct this algorithm using lattice basis reduction!

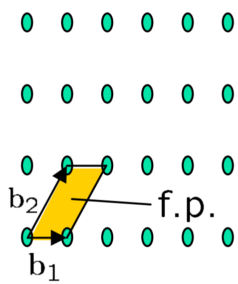
Point Lattices (Brief Intro)



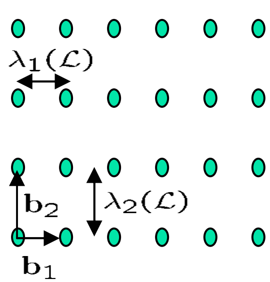
- Definition (Lattice): Given a basis of n linearly-independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in vector space $\mathbb{R}^n$, we call the set $\mathcal{L}$ of all integer linear combinations of these vectors a lattice of dimension n



- A basis matrix B of lattice $\mathcal{L}$ is an $n \times n$ matrix listing basis vectors in rows
- The determinant $\det(\mathcal{L})$ of lattice $\mathcal{L}$ is $|\det(B)|$ where B is any basis matrix for $\mathcal{L}$.
  - Geometrically, $\det(\mathcal{L})$ is equal to the volume of any fundamental parallelpiped (f.p.) of .



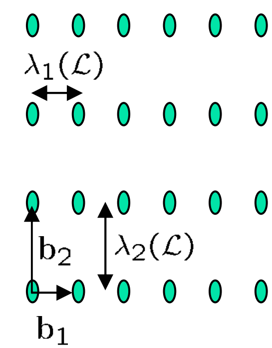
- We use infinity-norm $\|\cdot\|_\infty$(max. abs. value of coordinates) to measure "length" of lattice vectors
- Define "Minkowski Minima" $\lambda_1(\mathcal{L}), \ldots, \lambda_n(\mathcal{L})$ of lattice $\mathcal{L}$:
  - $\lambda_1(\mathcal{L})$ = shortest infinity-norm over all non-zero vectors of $\mathcal{L}$
  - $\lambda_i(\mathcal{L})$ = shortest infinity-norm bound over all i linearly-independent vectors of $\mathcal{L}$



Point Lattices (Brief Intro)

Theorem 1 (Minkowski's First Theorem) *Let $\mathcal{L}$ be a lattice in $\mathbb{R}^n$. Then*

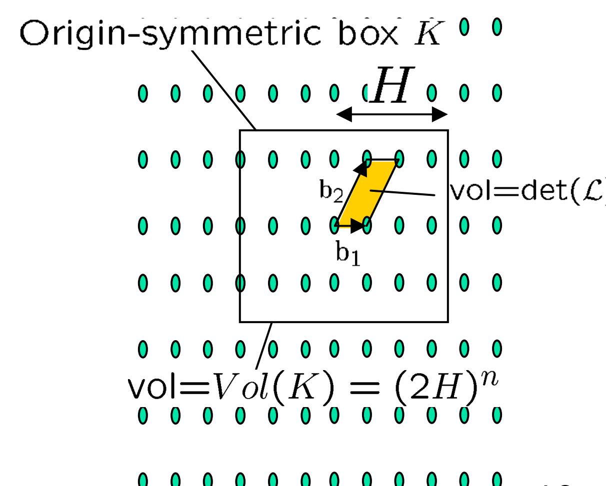
$$\lambda_1(\mathcal{L}) \leq \det(\mathcal{L})^{\frac{1}{n}}.$$



Theorem 2 (Minkowski's Second Theorem) *Let $\mathcal{L}$ be a lattice in $\mathbb{R}^n$. Then*

$$(\lambda_1(\mathcal{L}) \cdots \lambda_n(\mathcal{L}))^{1/n} \leq 2 \det(\mathcal{L})^{1/n}.$$

Theorem.[Blichfeldt-Corput] Let $\mathcal{L}$ be a lattice in $\mathbb{R}^n$ and let K denote the origin-centered box $\{\mathbf{v} \in \mathbb{R}^n : \|\mathbf{v}\|_\infty < H\}$ of volume $Vol(K) = (2H)^n$. Then the number of points of the lattice $\mathcal{L}$ contained in the box K is at least $2 \cdot Int\left(\frac{Vol(K)}{2^n \det(\mathcal{L})}\right) + 1$, where for any $z \in \mathbb{R}$, $Int(z)$ denotes the largest integer which is strictly less than z.



Point Lattices (Brief Intro)

- The Closest Vector Problem (CVP)

  Given a basis for a lattice $\mathcal{L}$ in $\mathbb{Q}^n$, and a "target" vector $\mathbf{t} \in \mathbb{Q}^n$, find a closest lattice vector $\mathbf{v} \in \mathcal{L}$ (i.e. $\|\mathbf{v}-\mathbf{t}\|_\infty = \min_{\mathbf{u}\in\mathcal{L}} \|\mathbf{u}-\mathbf{t}\|_\infty$).
- Exact (and near-exact) version of CVP is hard to solve efficiently in theory (NP-hard)
- But efficient Approximate-CVP algorithms exist

  An algorithm is called a *CVP approximation algorithm with* $\|\cdot\|_\infty$*-approximation factor* γ_{CVP} if it is guaranteed to find a lattice vector $\mathbf{v}$ such that $\|\mathbf{v}-\mathbf{t}\|_\infty \le \gamma_{CVP} \cdot \min_{\mathbf{u}\in\mathcal{L}} \|\mathbf{u}-\mathbf{t}\|_\infty$.
- First polynomial-time algorithm [Babai '86] suffices for us:

$$\gamma_{Bab} = n^{1/2} 2^{n/2}$$

Threshold-Changeability for Classical Shamir Scheme - Algorithms

- Increasing the threshold from t to t' > t

We use an efficient CVP approx. algorithm $\mathsf{A}_{\mathsf{CVP}}$ with approx. factor γ_{CVP}. Let $\Gamma_{CVP} = \log(\lceil \gamma_{CVP} + 1 \rceil)$ ($= O(t' + t)$ for Babai).

$\mathsf{H}_i(s_i)$ (ith Subshare Generation): To transform share $s_i \in \mathbf{Z}_p$ of original (t, n)-threshold scheme into subshare $t_i \in \mathbf{Z}_p$ of desired (t', n)-threshold scheme ($t' > t$) the ith shareholder does the following (for all $i = 1, \ldots, n$):

1 Determine noise bound H for δ_c-correctness

  (a) Set $H = \max(\lfloor p^{\alpha}/2 \rfloor, 1)$ with

  (b) $\alpha = 1 - \frac{1+\delta_F}{(t'/t)} > 0$ (noise bitlength fraction)

  (c) $\delta_F = \frac{(t'/t)}{k} \left(\log(\delta_c^{-1/t'} nt) + \Gamma_{CVP} + 1 \right)$.

2 Compute $t_i = \alpha_i \cdot s_i + r_i \bmod p$ for a uniformly random integer r_i with $|r_i| < H$.

Threshold-Changeability for Classical Shamir Scheme - Algorithms

- Noisy subshares decoding algorithm (subshare combiner)

$C'_{k,x}(\mathbf{t}_I)$ (Subshare Combiner): To combine subshares $\mathbf{t}_I = (t_i : i \in I)$ for some $I = \{i[1], \ldots, i[t']\}$ with $\#I = t'$ (for δ_c-correctness):

1. Build the following $(t'+t) \times (t'+t)$ matrix $M_{Sha}(\alpha_I, H, p)$, whose rows form a basis for a full-rank lattice $\mathcal{L}_{Sha}(\alpha_I, H, p)$ in $\mathbb{Q}^{t'+t}$:

$$M_{Sha}(\alpha_I, H, p) = \begin{pmatrix} p & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & p & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & p & 0 & 0 & \ldots & 0 \\ \alpha_{i[1]} & \alpha_{i[2]} & \ldots & \alpha_{i[t']} & H/p & 0 & \ldots & 0 \\ \alpha^2_{i[1]} & \alpha^2_{i[2]} & \ldots & \alpha^2_{i[t']} & 0 & H/p & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha^t_{i[1]} & \alpha^t_{i[2]} & \ldots & \alpha^t_{i[t']} & 0 & 0 & \ldots & H/p \end{pmatrix}.$$

Here $H = \lfloor p^{\alpha}/2 \rfloor$, $\alpha = 1 - \frac{1+\delta_F}{(t'/t)}$, $\delta_F = \frac{(t'/t)}{k}\left(\log(\delta_c^{-1/t'} nt) + \Gamma_{CVP} + 1\right)$.

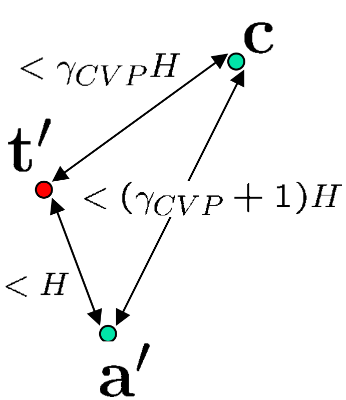
2. Define $\mathbf{t}' = (t_{i[1]}, \ldots, t_{i[t']}, 0, 0, \ldots, 0) \in \mathbf{Z}^{t'+t}$.

3. Run CVP Approx. alg. $\mathsf{A}_{\mathsf{CVP}}$ on lattice $\mathcal{L}_{Sha}(\alpha_I, H, p)$ with target vector $\mathbf{t}'$. Let $\mathbf{c} = (c_1, \ldots, c_{t'}, c_{t'+1}, \ldots, c_{t'+t}) \in \mathbb{Q}^{t'+t}$ denote the output vector returned by $\mathsf{A}_{\mathsf{CVP}}$.

4. Compute recovered secret $\widehat{s} = (p/H) \cdot c_{t'+1} \bmod p$.

Threshold-Changeability for Classical Shamir Scheme - Correctness

- Decoding algorithm correctness analysis (Main ideas):
  - By construction, the dealer's secret polynomial
    $$a(x) = s + a_1 x + \cdots + a_{t-1} x^{t-1}$$
  - gives rise to a lattice vector
    $$\mathbf{a}' = (\alpha_{i[1]} a(\alpha_{i[1]}) - k_1 p, \ldots, \alpha_{i[t']} a(\alpha_{i[t']}) - k_{t'} p, \frac{s}{p} H, \frac{a_1}{p} H, \ldots, \frac{a_{t-1}}{p} H)$$
  - which is "close" to the target vector
    $$\mathbf{t}' = (\alpha_{i[1]} a(\alpha_{i[1]}) - k_1 p + r_{i[1]}, \ldots, \alpha_{i[t']} a(\alpha_{i[t']}) - k_{t'} p + r_{i[t']}, 0, 0, \ldots, 0)$$
  - That is, $\|\mathbf{a}' - \mathbf{t}'\|_\infty < H$, so the approx. "close" lattice vector $\mathbf{c}$ returned by A_{CVP} satisfies $\|\mathbf{c} - \mathbf{t}'\|_\infty < \gamma_{CVP} H$.
  - By triangle inequality, the "error" lattice vector $\mathbf{z} = \mathbf{c} - \mathbf{a}'$ is "short" : $\|\mathbf{z}\|_\infty < (\gamma + 1) H$
  - and our algorithm fails only if this "error" lattice vector is "bad" in the sense: $\frac{p}{H}\mathbf{c}[t'+1] - \frac{p}{H}\mathbf{a}'[t'+1] = \frac{p}{H}\mathbf{z}[t'+1] \not\equiv 0 \pmod{p}$
  - We use counting argument to upper bound number of public vectors α_I for which $\mathcal{L}_{Sha}(\alpha_I)$ contains "short" and "bad" vectors



Threshold-Changeability for Classical Shamir Scheme - Correctness

- Algorithm correctness analysis (continued)
  - Counting argument to upper bound number of public vectors α_I for which $\mathcal{L}_{Sha}(\alpha_I)$ contains "short" and "bad" vectors reduces to following algebraic counting lemma:

    Lemma. Fix a prime p, positive integers (n, t, H), and a non-empty set A of polynomials over $\mathbf{Z}_p$ of degree at least 1 and at most t. The number of vectors $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbf{Z}_p^n$ for which there exists a polynomial $a \in A$ such that $\|a(\alpha_i)\|_{L,p} < H$ for all $i = 1, \ldots, n$ is upper bounded by $\#A \cdot (2Ht)^n$.
  - We use this to obtain an upper bound on fraction of "bad" public vectors $(\alpha_1, \ldots, \alpha_n) \in (\mathbf{Z}_p)^n$ for which combiner may not always work
  - This "bad" fraction δ_c can be made as small as we wish, for sufficiently large security parameter $k = O(\log \delta_c^{-1})$

Threshold-Changeability for Classical Shamir Scheme - Security

- Security Analysis (Main Ideas):
  - We assume a uniform distribution on secret space $\mathbf{Z}_p$:
    - Secret entropy $H(s \in \mathbf{Z}_p) = \log p \in [k, k+1]$
  - We show that, for all choices of the public vector $\alpha_I \in D((\mathbf{Z}_p^*)^{t_s})$ except for a small "bad" fraction $\delta_s = O(1/k^{t'})$, the following holds:
  - For all subshare subsets $I \subseteq [n]$ of size $\#I = t_s \leq Int(f(k)(t' - t'/t))$ with $\lim_{k \to \infty} f(k) = 1$
  - and all values $\mathbf{s}_I = (s_{i[1]}, \ldots, s_{i[t_s]})$ for the corresponding subshare vector,
  - the conditional probability distribution $P_{k,x}(\cdot|\mathbf{s}_I)$ for the secret given the observed subshare vector value $\mathbf{s}_I$ is "close" to uniform: $P_{k,x}(s|\mathbf{s}_I) \leq 2^{\epsilon_s}/p$ for all $s \in \mathbf{Z}_p$ with $\epsilon_s(k) = O(\log k)$
  - → Secret <u>entropy loss</u> is bounded as (for all I and $\mathbf{s}_I$)

$$L_{k,x}(\mathbf{s}_I) = |H(s \in \mathbf{Z}_p) - H(s \in \mathbf{Z}_p|\mathbf{s}_I)| \leq \epsilon_s(k)$$

Threshold-Changeability for Classical Shamir Scheme - Security

- Security analysis (cont.)
  - To derive bound $P_{k,x}(s|\mathbf{s}_I) \le 2^{\epsilon_s}/p$ for all $s \in \mathbf{Z}_p$ we observe
$$P_{k,x}(s|\mathbf{s}_I) = \frac{\#S_{s,p}(\alpha_I, t, p, H, \mathbf{s}_I)}{\#S_{0,1}(\alpha_I, t, p, H, \mathbf{s}_I)},$$
  - where for integers $\widehat{s} \in \{0, s\}$ and $\widehat{p} \in \{1, p\}$ we define

$$S_{\widehat{s},\widehat{p}}(\alpha_I, t, p, H, \mathbf{s}_I) \stackrel{\text{def}}{=} \{a \in \mathbf{Z}_p[x; t-1] : \|\alpha_{i[j]} a(\alpha_{i[j]}) - s_{i[j]}\|_{L,p} < H \forall j \in [t_s] \text{ and } a(0) \equiv \widehat{s} \pmod{\widehat{p}}\}.$$

  - We lower bound $\#S_{0,1}$ (no. of dealer poly consistent with shares)
  - We upper bound $\#S_{s,p}$ (no. of dealer poly consistent with shares and any fixed value s for the secret)

Threshold-Changeability for Classical Shamir Scheme - Security

- Security analysis (cont.)
  - We first reduce the problem to lattice point counting:

Lemma. Let $\mathcal{L}_{Sha}(\alpha_I, t, p, H, \widehat{p})$ denote the lattice with basis matrix

$$M_{Sha}(\alpha_I, t, p, H, \widehat{p}) = \begin{pmatrix} p & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & p & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & p & 0 & 0 & \dots & 0 \\ \widehat{p}\alpha_{i[1]} & \widehat{p}\alpha_{i[2]} & \dots & \widehat{p}\alpha_{i[t_s]} & 2H/(p/\widehat{p}) & 0 & \dots & 0 \\ \alpha_{i[1]}^2 & \alpha_{i[2]}^2 & \dots & \alpha_{i[t_s]}^2 & 0 & 2H/p & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{i[1]}^t & \alpha_{i[2]}^t & \dots & \alpha_{i[t_s]}^t & 0 & 0 & \dots & 2H/p \end{pmatrix} \begin{array}{l} \leftarrow \mathbf{b}_1 \\ \\ \vdots \\ \\ \\ \\ \\ \leftarrow \mathbf{b}_{t_s+t} \end{array},$$

and define the vector $\widehat{\mathbf{s}}_I \in \mathbb{Q}_{t_s+t}$ by

$$\widehat{\mathbf{s}}_I \stackrel{\text{def}}{=} \left(s_{i[1]} - \widehat{s}\alpha_{i[1]}, \dots, s_{i[t_s]} - \widehat{s}\alpha_{i[t_s]}, H(1 - \frac{1+2\widehat{s}}{p}), H(1-\frac{1}{p}), \dots, H(1-\frac{1}{p})\right).$$

Then the sizes of the following two sets are equal:

$$S_{\widehat{s},\widehat{p}}(\alpha_I, t, p, H, \mathbf{s}_I) \stackrel{\text{def}}{=} \{a \in \mathbf{Z}_p[x; t-1] : \|\alpha_{i[j]}a(\alpha_{i[j]}) - s_{i[j]}\|_{L,p} < H \forall j \in [t_s] \text{ and } a(0) \equiv \widehat{s} \pmod{\widehat{p}}\},$$

and

$$V_{\widehat{s},\widehat{p}}(\alpha_I, t, p, H, \widehat{\mathbf{s}}_I) \stackrel{\text{def}}{=} \{\mathbf{v} \in \mathcal{L}_{Sha}(\alpha_I, t, p, H, \widehat{p}) : \|\mathbf{v} - \widehat{\mathbf{s}}_I\|_\infty < H\}.$$

Proof idea: We define a 1-1 and onto map from $V_{\widehat{s},\widehat{p}}$ to $S_{\widehat{s},\widehat{p}}$ by mapping vector

$$\mathbf{v} = k_1^{\mathbf{v}}\mathbf{b}_1 + \dots k_{t_s}^{\mathbf{v}}\mathbf{b}_{t_s} + k^{\mathbf{v}}\mathbf{b}_{t_s+1} + a_1^{\mathbf{v}}\mathbf{b}_{t_s+2} + \dots + a_{t-1}^{\mathbf{v}}\mathbf{b}_{t_s+t}$$

to polynomial

$$a_{\mathbf{v}}(x) = \lfloor \widehat{s} + k^{\mathbf{v}}\widehat{p} \rfloor_p + \lfloor a_1^{\mathbf{v}} \rfloor_p x + \dots + \lfloor a_{t-1}^{\mathbf{v}} \rfloor_p x^{t-1}$$

Threshold-Changeability for Classical Shamir Scheme - Security

- Security analysis (cont.)
  - Now we use lattice tools to lower bound $\#V_{0,1}$
  - Note $\#V_{0,1}$ is a "non-homogenous" counting problem: we need the number of lattice points in a box $T_{\mathbf{s}_I}(H) = \{\mathbf{v} \in \mathbb{Q}^{t_s+t} : \|\mathbf{v} - \widehat{\mathbf{s}}_I\|_\infty < H\}$ centred on a (non-lattice) vector $\widehat{\mathbf{s}}_I$
  - We reduce this non-homogenous problem to two simpler problems:
    - The homogenous problem of lower bounding the number of lattice points in an origin-centred box

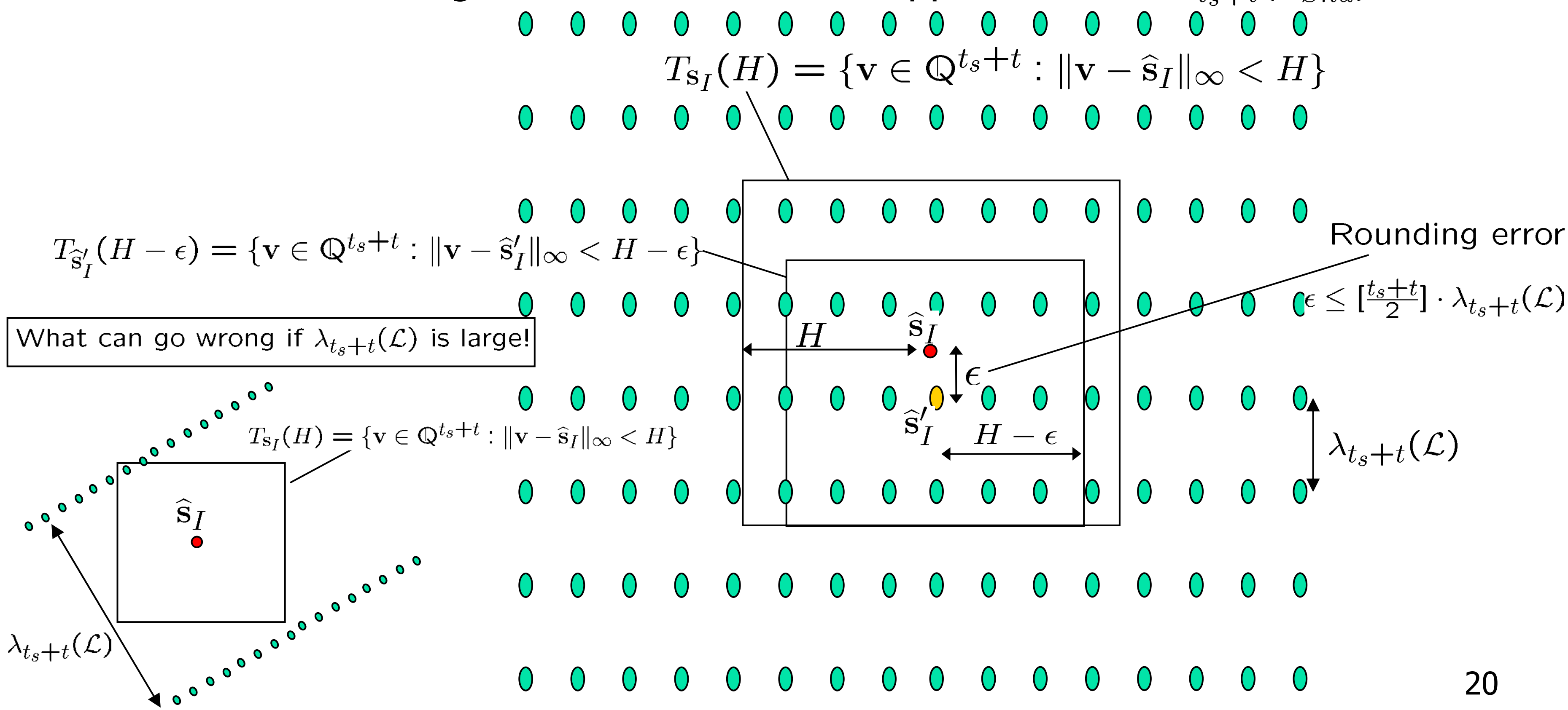
$$T_{\mathbf{0}}(H-\epsilon) = \{\mathbf{v} \in \mathbb{Q}^{t_s+t} : \|\mathbf{v}\|_\infty < H-\epsilon\} \quad \text{where } \epsilon \le \left(\frac{t_s+t}{2}\right)\lambda_{t_s+t}(\mathcal{L}_{Sha})$$

    - Upper bounding the largest Minkowski minimum $\lambda_{t_s+t}(\mathcal{L}_{Sha})$

$$\boxed{\text{We show } \#V_{\widehat{s},\widehat{p}_0} \ge \#\{\mathbf{v} \in T_{\mathbf{0}}(H-\epsilon) \cap \mathcal{L}_{CRT}\}}$$

Threshold-Changeability for Classical Shamir Scheme - Security

- Security analysis (cont.)
  - Proof idea of reduction of "non-homogenous lower bound" to "homogenous lower bound" + upper bound on $\lambda_{t_s+t}(\mathcal{L}_{Sha})$



Threshold-Changeability for Classical Shamir Scheme - Security

- Security analysis (cont.)
  - Problem 1 (point counting in origin-symmetric box) is solved directly by applying Blichfeldt-Corput Theorem:

$$\#\{\mathbf{v} \in \mathcal{L}_{Sha} \bigcap T_0(H-\epsilon)\} \geq 2Int\left(\frac{Vol(T_0(H-\epsilon))}{2^{t_s+t}\det(\mathcal{L}_{Sha})}\right)$$

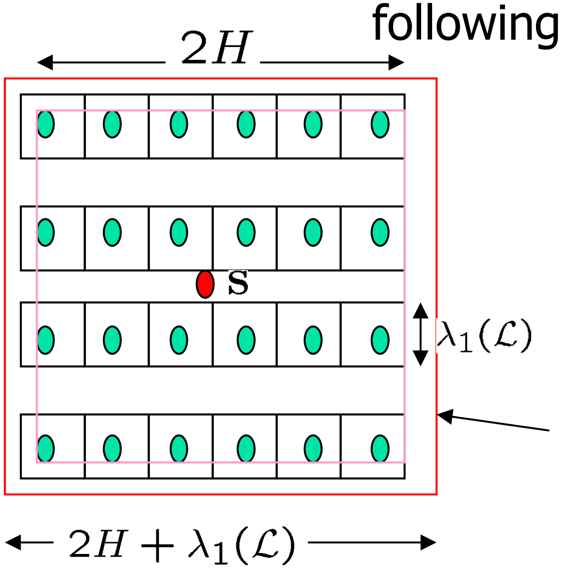
  - Problem 2 (upper bounding $\lambda_{t_s+t}(\mathcal{L}_{Sha})$) is solved by applying Minkowski's Second Theorem to reduce it first to the problem of <u>lower bounding the first Minkowski minimum</u>(shortest vector norm)

$$\lambda_{t_s+t}(\mathcal{L}_{Sha}) \leq \frac{2^{t_s+t}\det(\mathcal{L}_{Sha})}{\lambda_1(\mathcal{L}_{Sha})^{t_s+t-1}}$$

    - We lower bound the first Minkowski minimum $\lambda_1(\mathcal{L}_{Sha})$ (except for a "small" fraction of "bad" public vectors $(\alpha_1, \ldots, \alpha_n)$) by applying our algebraic counting lemma (using similar argument used in correctness analysis)

Threshold-Changeability for Classical Shamir Scheme - Security

- Security analysis (cont.)
  - This completes the results needed to lower bound $\#V_{0,1}$
  - Recall that we also need to upper bound $\#V_{s,p}$
  - We reduce this problem also to lower bounding $\lambda_1(\mathcal{L}_{Sha})$ with the following result:



Lemma. For any lattice $\mathcal{L}$ in $\mathbb{R}^n$, vector $\mathbf{s} \in \mathbb{R}^n$, and $H > 0$, we have

$$\#\{\mathbf{v} \in \mathcal{L} : \|\mathbf{v} - \mathbf{s}\|_\infty < H\} \leq \left[\frac{2H}{\lambda_1(\mathcal{L})} + 1\right]^n.$$

Upper bound total vol of small boxes $\#V \times \lambda_1^n$ by volume of large box $(2H + \lambda_1(\mathcal{L}))^n$

  - And now we use our lower bound on $\lambda_1(\mathcal{L}_{Sha})$ again!

Conclusions

- Presented lattice-based threshold changeability algorithms for Shamir secret-sharing
- Proved concrete bounds on correctness and security using classical results from theory of lattices