# IP-SAS: Preserving Incumbent Users' Privacy in Exclusion-Zone-Based Spectrum Access Systems

Yanzhi Dou∗, Kexiong (Curtis) Zeng∗, Yaling Yang∗, and Kui Ren†

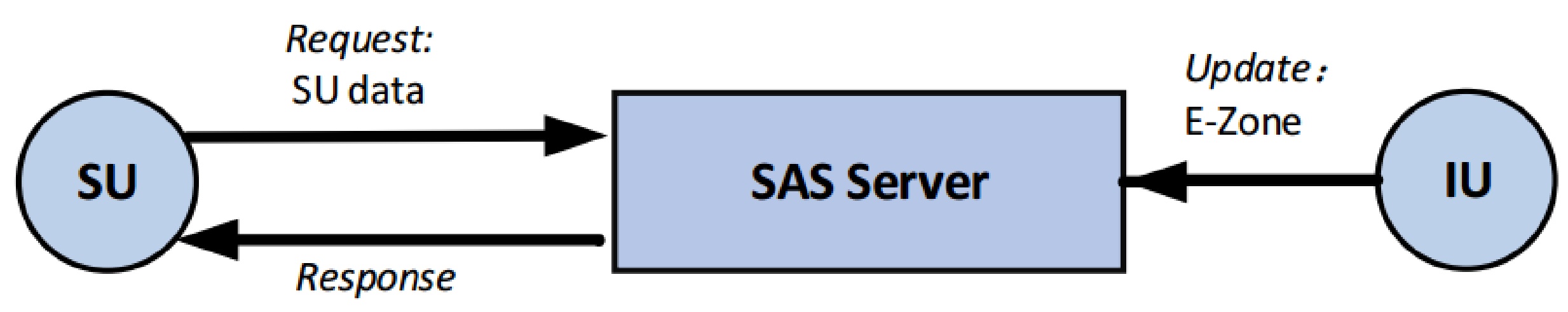
∗ Virginia Tech (USA)   †SUNY at Buffalo (USA)

## Background & Motivation

- To mitigate the potential spectrum scarcity problem, PCAST and FCC have proposed spectrum access system (SAS) to realize the full potential of government-hold spectrum by sharing with wireless broadband operators/users.
- In current SAS proposals, the sensitive operation information of federal incumbent users (IUs) needs to be shared with the centralized SAS to realize spectrum allocation.
- However, SAS is not necessarily trust-worthy for holding such sensitive IU data. Particularly, PCAST and FCC allow industry third parties (e.g., Google) to operate SAS to enhance its efficiency and scalability.
- Therefore, the current SAS proposals dissatisfy the IUs' privacy requirement.

## Problem Statement

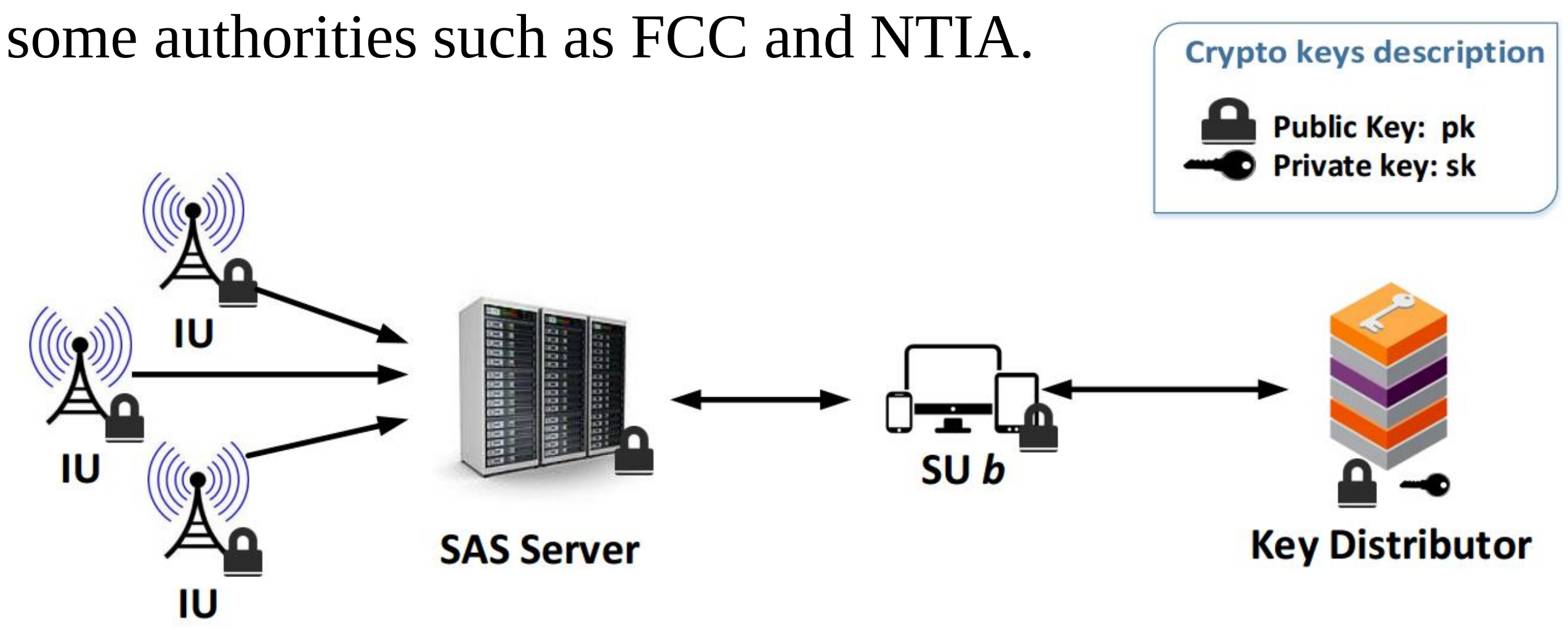
- **System Model:** In a typical scenario of exclusion-zone-based SAS systems, IUs first compute their exclusion zones (E-Zones) and send the E-Zone data to SAS in the initialization phase. When an SU wants to access the spectrum, it needs to provide its operation parameters and geolocation to SAS. SAS checks whether the SU is within the E-Zone of any IU. For a given spectrum, if the answer is yes (no), SAS denies (permits) the SU's spectrum access to this spectrum.



  - Remark: The privacy issue of the protection-zone-based SAS, where multiple SUs' interference will be aggregated, has been addressed in our MobiHoc paper [1].
- **Adversary Model:** Semi-honest model; Malicious model.
- **Design Goals:** Correctness; Privacy; Resistance to malicious attacks; Efficiency.

## Basic Design for semi-honest attack model

IP-SAS involves four parties: (1) a SAS Server *S* for spectrum allocation, (2) IUs, (3) SUs, and (4) a Key Distributor *K*. *K* creates a Paillier public/private key pair (pk, sk) and is trusted for keeping sk secret. In the real world, the role of *K* can be played by some authorities such as FCC and NTIA.



Protocol description:

**I. Initialization Phase:**

$\mathcal{K}$:

(1) $\mathcal{K}$ runs KeyGen and generates a Paillier key pair $(\mathsf{pk}, \mathsf{sk})$. pk is distributed to $\mathcal{S}$ and IUs, and sk is kept secret.

*IUs* (numbered as $1, 2, \ldots, k, \ldots, K$):

(2) IU $k$ calculates its E-Zone map $\mathbf{T}_k$.

(3) IU $k$ encrypts $\mathbf{T}_k$ with pk and gets $\widehat{\mathbf{T}}_k$.

(4) IU $k$ uploads $\widehat{\mathbf{T}}_k$ to $\mathcal{S}$.

$\mathcal{S}$:

(5) Upon all IUs having uploaded their E-Zone map, $\mathcal{S}$ computes $\widehat{\mathbf{M}} := \oplus_{k \in \{1,2,\ldots,K\}} \widehat{\mathbf{T}}_k$ to aggregate the E-Zone map of all IUs and generates a global E-Zone map $\widehat{\mathbf{M}}$.

**II. Spectrum Computation Phase:**

*SU b:*

(6) SU $b$ submits spectrum request containing its operation parameters $(h_s, p_{ts}, g_{rs}, i_s)$ and location $l$ to $\mathcal{S}$.

$\mathcal{S}$:

(7) $\mathcal{S}$ retrieves the corresponding entry in the global E-Zone map $\widehat{\mathbf{M}}$ and obtains $\widehat{\mathbf{X}}_b$.

(8) $\mathcal{S}$ adds random blinding factor $\widehat{\beta}$ to $\widehat{\mathbf{X}}_b$ to generate $\widehat{\mathbf{Y}}_b$.

(9) $\mathcal{S}$ returns $\widehat{\mathbf{Y}}_b$ and $\beta$ to SU $b$.

**III. Recovery Phase:**

*SU b:*

(10) SU $b$ relays $\widehat{\mathbf{Y}}_b$ to $\mathcal{K}$ for decryption.

$\mathcal{K}$:

(11) $\mathcal{K}$ decrypts $\widehat{\mathbf{Y}}_b$ with sk and returns $\mathbf{Y}_b$ to SU $b$.

*SU b:*

(12) SU $b$ recovers $\mathbf{X}_b$ by removing the blinding factor $\beta$ from $\mathbf{Y}_b$.

Notes:

1. $T_k(l, f, h_s, p_{ts}, g_{rs}, i_s) := \begin{cases} \epsilon, & \text{grid } l \in EZ(f, h_s, p_{ts}, g_{rs}, i_s) \\ 0, & \text{grid } l \notin EZ(f, h_s, p_{ts}, g_{rs}, i_s) \end{cases}$

2. Why Paillier Cryptosystem? Because it supports homomorphic additions.

**Homomorphic Addition** $(\mathsf{Add}_{\mathsf{pk}})$:
$\mathsf{Dec}_{\mathsf{sk}}\left(\mathsf{Add}_{\mathsf{pk}}(\widehat{m_1}, \widehat{m_2})\right) = \mathsf{Dec}_{\mathsf{sk}}\left(\widehat{m_1} \cdot \widehat{m_2}\right) = m_1 + m_2.$

$\widehat{m}$ is used to denote the ciphertext of some message m.

## Countering Malicious Behaviors

Malicious behaviors are non-trivial to detect from data discrepancy due to the privacy-preserving feature of IP-SAS. We assume SUs and *S* are malicious.

- **Malicious SU**

➢ Attacks

Step 6: Submits faked operation data.

Step 12: Claims a different $X_b$'.

These attacks help SU gain illegal benefits of spectrum access.

➢ Countermeasures

Step 6: digital signature.

Step 12: digital signature + re-encryption

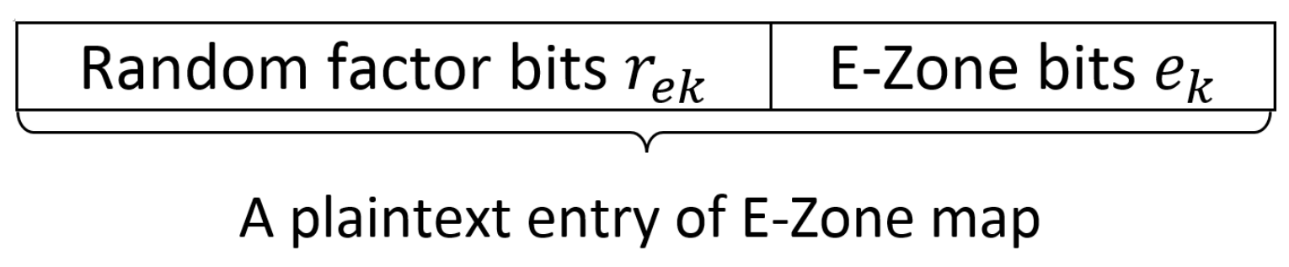
- **Malicious *S***

➢ Attacks

Step 5, 7, 8, 9: Deviation from these steps would make SU *b* recover a wrong $X_b$'.

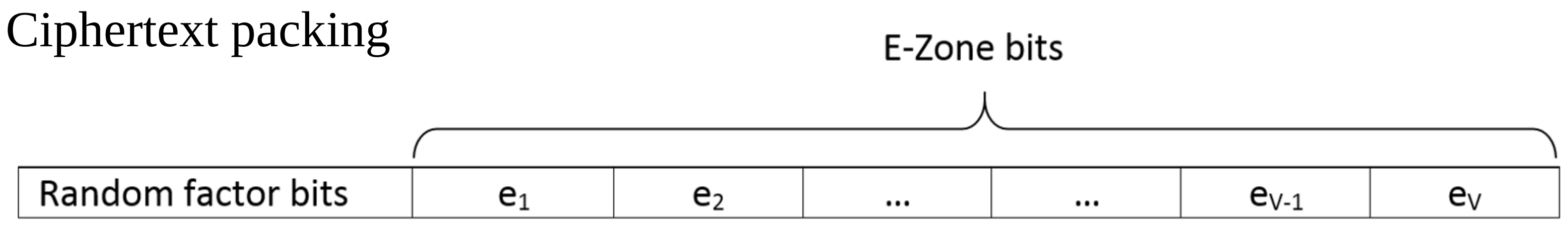
➢ Countermeasures

Zero-knowledge proof based on additive-homomorphic commitment (Pederson commitment scheme).



## Improving IP-SAS Efficiency

- Ciphertext packing



- Parallel computing: Distribute computation task to multiple servers.

## Evaluation Results

- Service area of IP-SAS: Washington D.C. (154.82 km$^2$).
- Radio propagation model to compute E-Zone: Longley-Rice model fed by high resolution terrain data SRTM3.
- 2048-bit Paillier implementation: 112-bit security level.
- Experiment parameter settings

| | |
|---|---|
| Number of IUs ($K$) | 500 |
| Number of grids ($L$) | 15482 |
| Number of frequency channels ($F$) | 10 |
| Number of SU antenna heights ($H_s$) | 5 |
| Number of SU effective radiated power values ($P_{ts}$) | 3 |
| Number of SU receiver antenna gain values ($G_{rs}$) | 3 |
| Number of SU interference tolerance thresholds ($I_s$) | 3 |

- Evaluation results: 1.25 seconds response time, 17.8 KB computation overhead for each SU spectrum request.

| | Before Acceleration | After Acceleration |
|---|---|---|
| (2) E-Zone map calculation | 21.2 hours | 1.65 hours |
| (3) Commitment | 11.7 hours | 3.21 minutes |
| (4) Encryption | 68.5 hours | 17.9 minutes |
| (6) Aggregation | 29.0 hours | 5.2 minutes |
| (8)-(10) $\mathcal{S}$ Response | 1.12 seconds | 1.11 seconds |
| (12)(13) Decryption | 0.134 seconds | 0.134 seconds |
| (15) Recovery | - | - |
| (16) Verification | 0.118 seconds | 0.118 seconds |

| | Before Packing | After Packing |
|---|---|---|
| (4) IU→ $\mathcal{S}$ | 9.97 GB | 510 MB |
| (6) SU→ $\mathcal{S}$ | 25 B | 25 B |
| (9) $\mathcal{S}$ →SU | 7.75 KB | 7.75 KB |
| (10) SU→ $\mathcal{K}$ | 5 KB | 5 KB |
| (13) $\mathcal{K}$ →SU | 5 KB | 5 KB |

## Conclusion

In this paper, we build IP-SAS to perform efficient SAS process while preserving IUs' privacy. This system takes advantage of additive-homomorphic encryption to allow secure SAS operations. Moreover, we design mechanisms to prevent malicious parties from compromising IP-SAS. Finally, we implement acceleration methods to increase IP-SAS's efficiency. Experiments based on real-world data demonstrate the scalability and practicality of IP-SAS in real-world deployment.

## References

[1] Y. Dou, K. C. Zeng, H. Li, Y. Yang, B. Gao, C. Guan, and S. Li, "P2-SAS: Preserving Users' Privacy in Centralized Dynamic Spectrum Access Systems," *in Proceedings of the 17th ACM MobiHoc*, 2016.

## Acknowledgement & Contact

This work is supported by US National Science Foundation under grant CNS-1547366, CNS-1228903, CNS-1054697.

For more information, please contact Yanzhi Dou (yzdou@vt.edu) or Yaling Yang (yyang8@vt.edu).