The 2nd Workshop on CPS & IoT Security and Privacy (CPSIoTSec '21)

Kyuin Lee University of Wisconsin–Madison
Younghyun Kim University of Wisconsin–Madison

Balancing Security and Usability of Zero-interaction Pairing and Authentication for the Internet-of-Things

November 15, 2021

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Secure Authentication

Nearly all IoT traffic is unencrypted

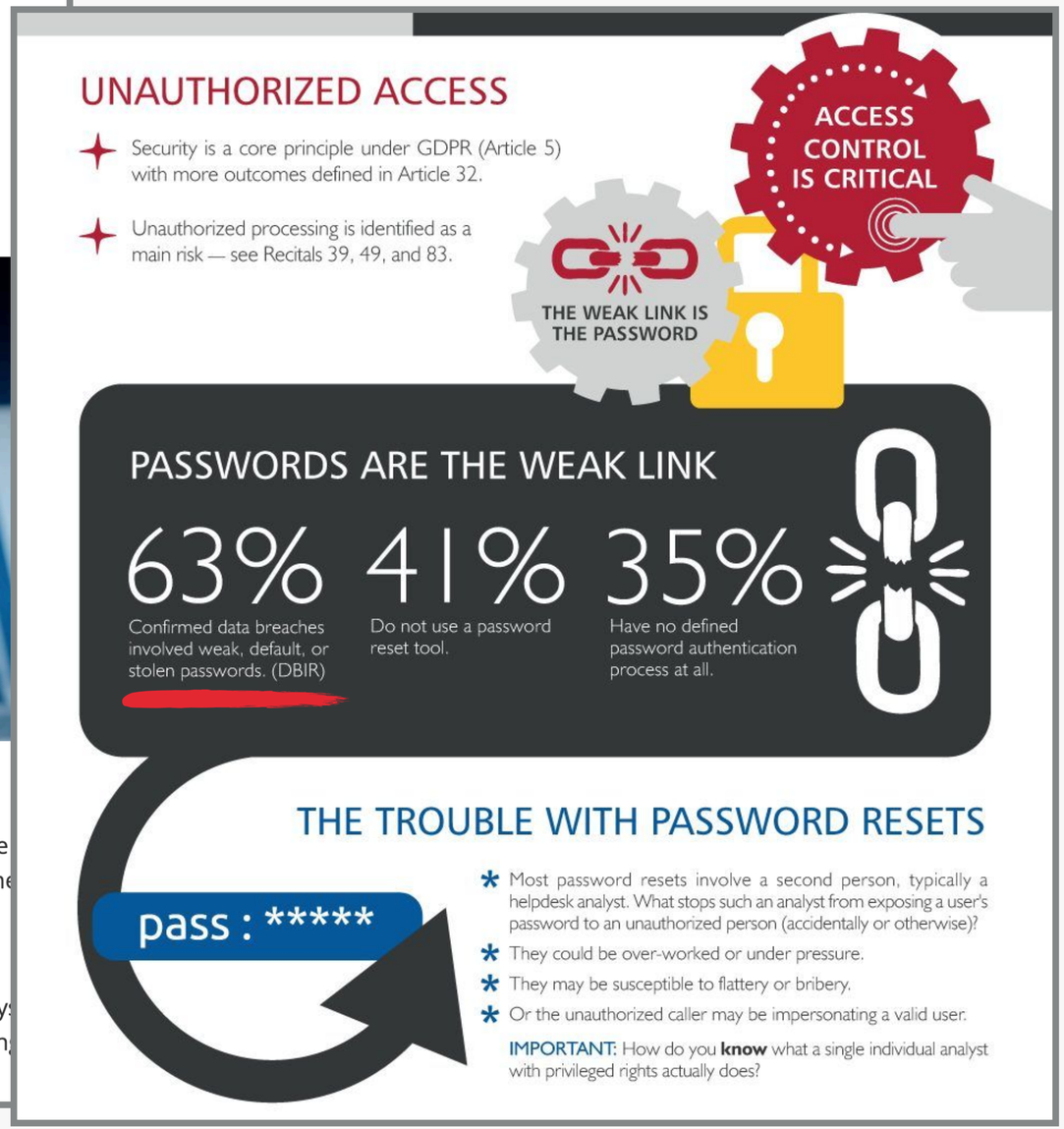
By Sead Fadilpašić a month ago

IoT devices are considered "low-hanging fruit" among cybercriminals.

(Image credit: Image source: Shutterstock/everything possible)

Practically all of the traffic flowing from Internet of Things (IoT) devices in the United States is not encrypted, consequently putting both businesses and the customers at unnecessary risk of data theft and all others that follow.

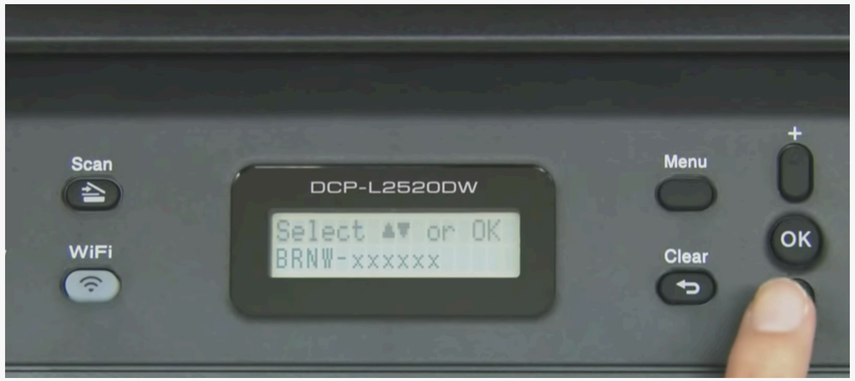
This is according to a new report by Unit 42, Pao Alto Networks' threat intelligence team, which analysed 1.2 million IoT devices in thousands of phys locations across enterprise IT and healthcare organisations in the U.S., findin that 98 per cent of all IoT device traffic is unencrypted.



Make your Service Desk GDPR Compliant
https://www.fastpasscorp.com/why-fastpass/insights/idc-gdpr-password-reset/

Usability Challenge in Current IoT Device Authentication

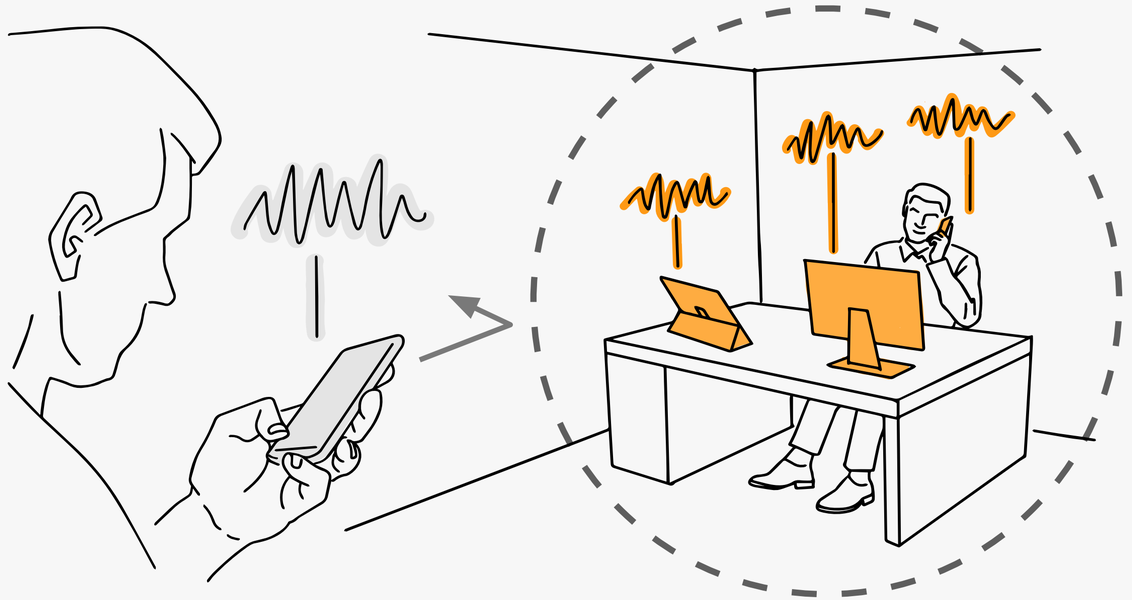
Initial authentication



All connections secured by user or pre-defined password

High user effort during initial authentication procedure

Zero-interaction Pairing and Authentication (ZIPA)

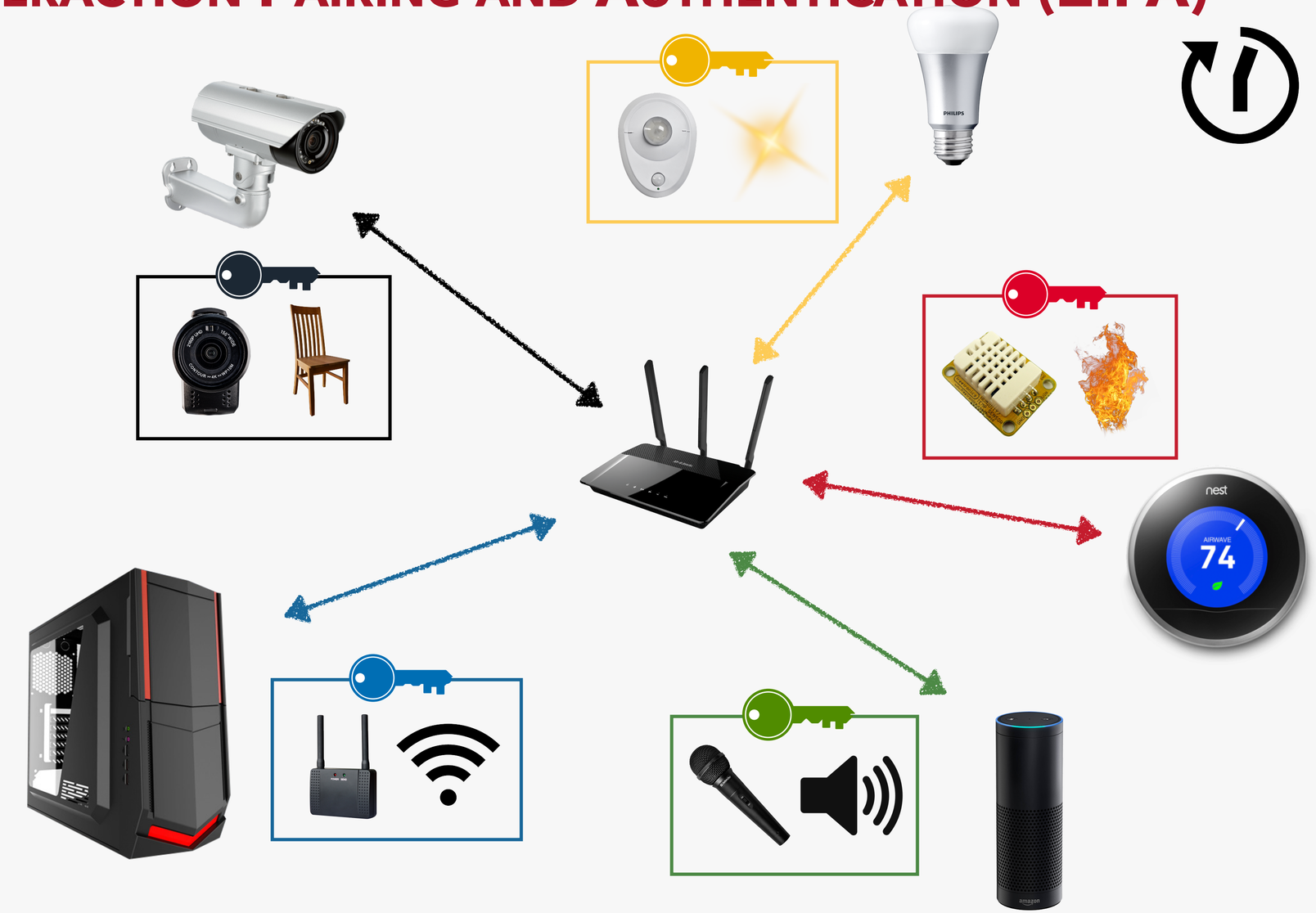


Adversary **Authentication range**

- Devices within authentication range under full control of legitimate user
- Adversarial devices cannot be placed within authentication range

Authentication range decides authentication or rejection of devices

Zero-interaction Pairing and Authentication (ZIPA)



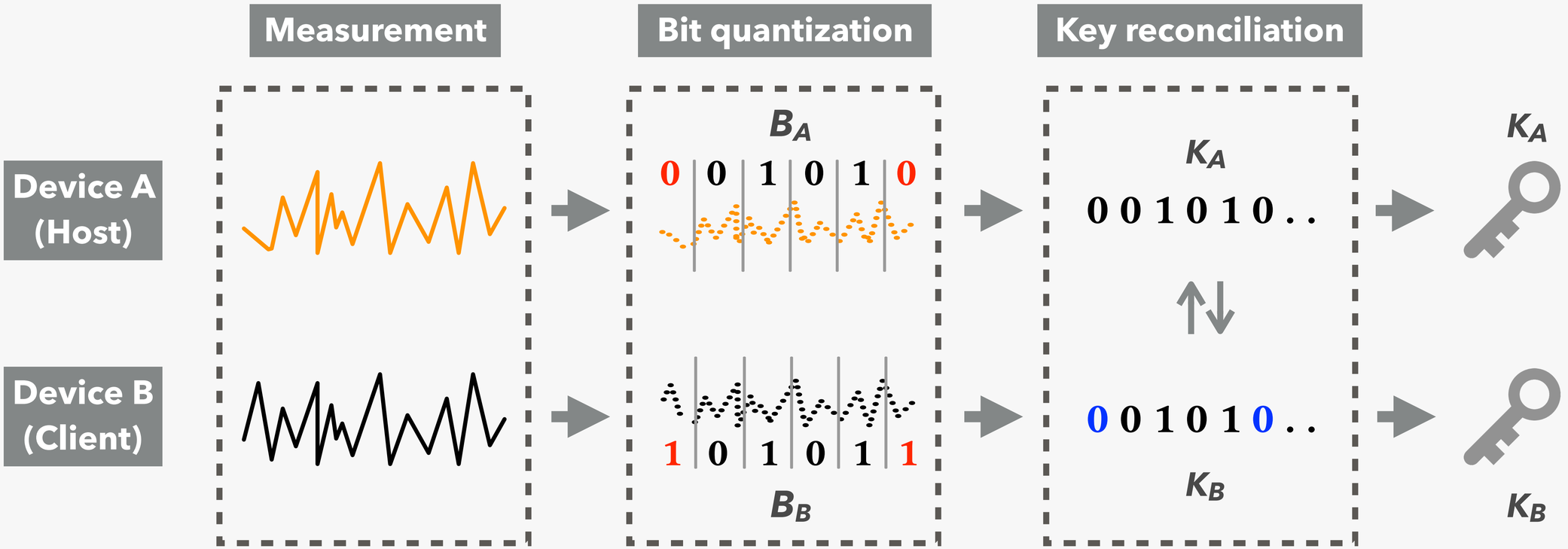
Periodic update

Zero-involvement, periodic update, diverse password

Research Questions

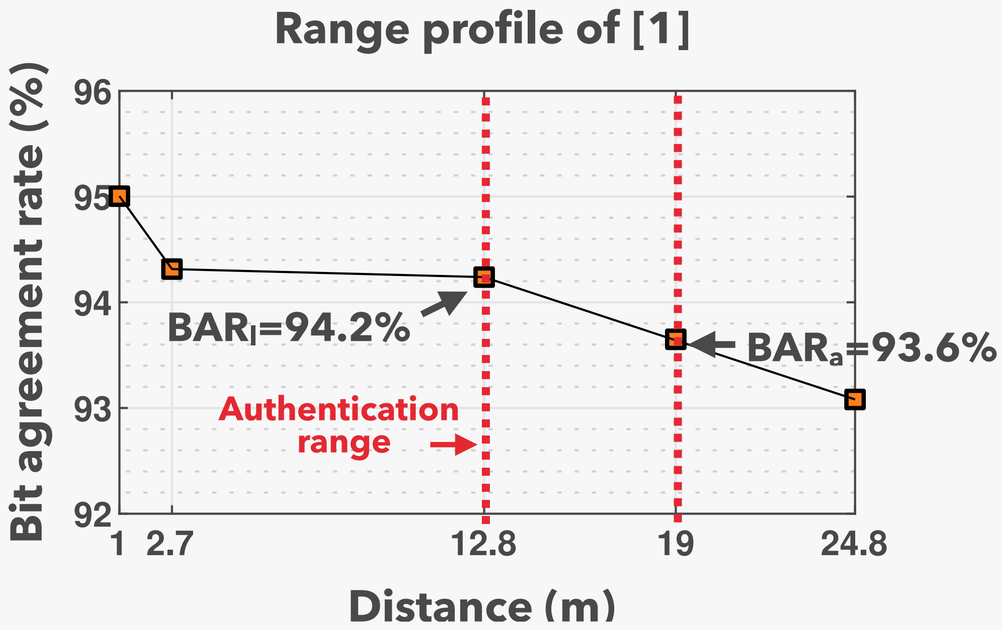
- **How can we easily balance security and usability of ZIPA methods?**
  - Key reconciliation parameter is an important factor to balance security vs. usability
  - We propose a generic framework to quickly obtain a balanced reconciliation parameter based on provided authentication range
- **Among many, which key reconciliation scheme should be used?**
  - Previous ZIPA works utilize different reconciliation schemes
  - We analyze two of the most widely used ones in terms of:
    - Error correcting performance
    - Entropy loss
    - Computation

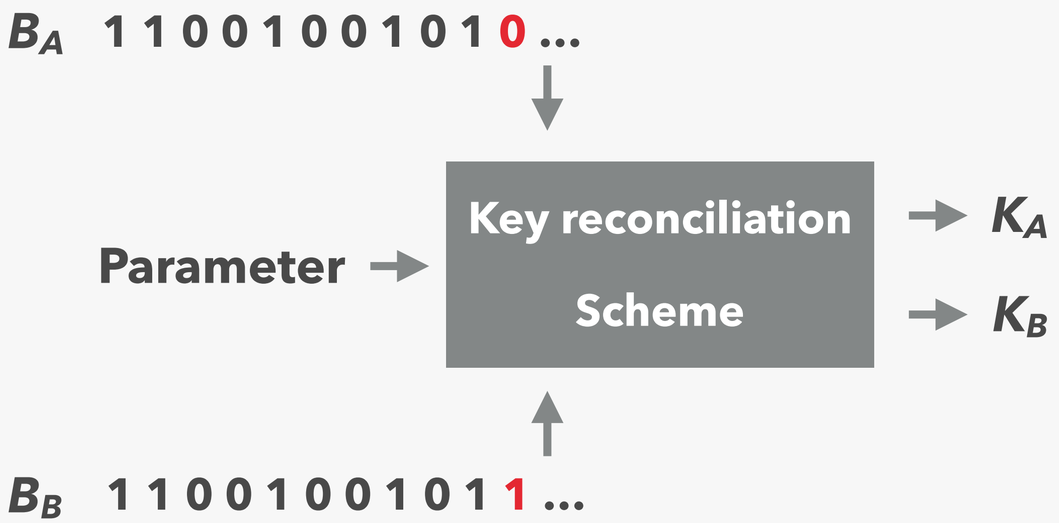
ZIPA's Three-Stage Pipeline



- **Measurement:** devices independently measure context signal
- **Bit quantization:** signal is quantized into environmental bits, *B*
- **Key reconciliation:** difference is corrected to final key, *K*

KEY RECONCILIATION



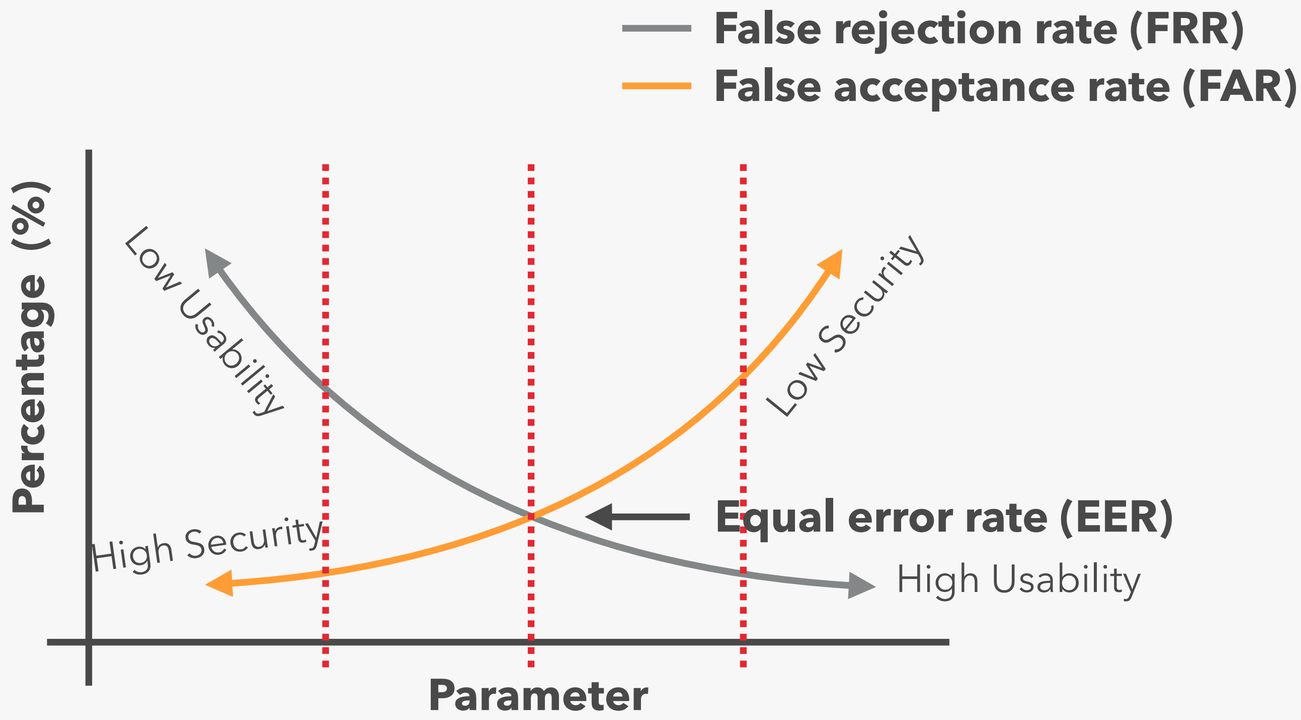


- **Bit agreement rate (BAR) decreases with increasing distance**
- **BAR$_l$:** lowest BAR achieved between legitimate device pairs
- **BAR$_a$:** highest BAR achieved between adversarial device pairs

The parameter of key reconciliation scheme determines bit correction

[1] VoltKey: Continuous Secret Key Generation Based on Power Line Noise for Zero-Involvement Pairing and Authentication
Kyuin Lee, Neil Klingensmith, Suman Banerjee, and Younghyun Kim
Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 3, 3, Article 93 (September 2019)

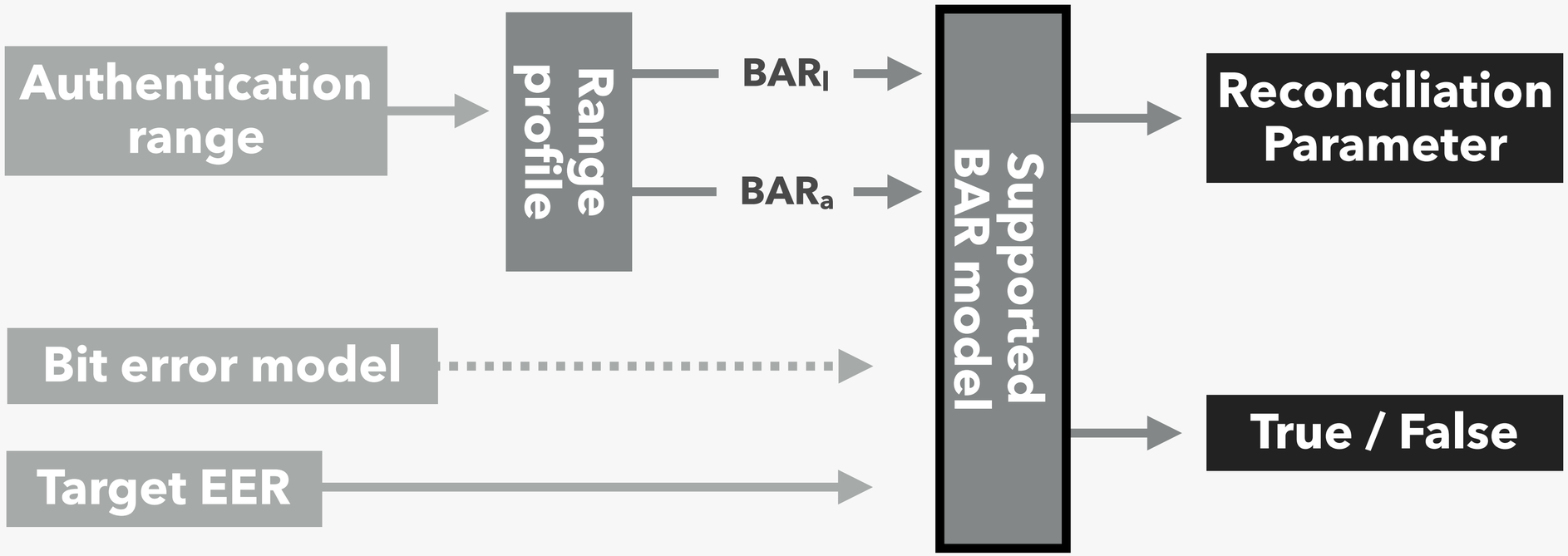
Balancing Security and Usability



- Too high error correcting parameter results in low usability
- Too low error correcting parameter results in low security

Reconciliation parameter needs to be on point of equal error rate (EER)

Proposed Framework



Supported BAR model can be built with different reconciliation schemes

KEY RECONCILIATION SCHEME (ECC–BASED, FUZZY COMMITMENT)

Device *A* (Host) | **Device *B* (Client)**

1) K_A = PRNG(k)

2) λ_A = RS.ENCODE(T, K_A)

3) $\sigma = B_A \oplus \lambda_A$

4) σ, SHA256(K_A) →

← $\lambda_B = \sigma \oplus B_B$

5) ← K_B = RS.DECODE(T, λ_A)

6) ← HASH(K_A) = HASH(K_B)

- k = length of K in bits
- PRNG = pseudo-random number generator
- ENCODE and DECODE = error correcting codes (i.e., Reed-Solomon (T,k))
- T is the error correction parameter during ENCODE and DECODE

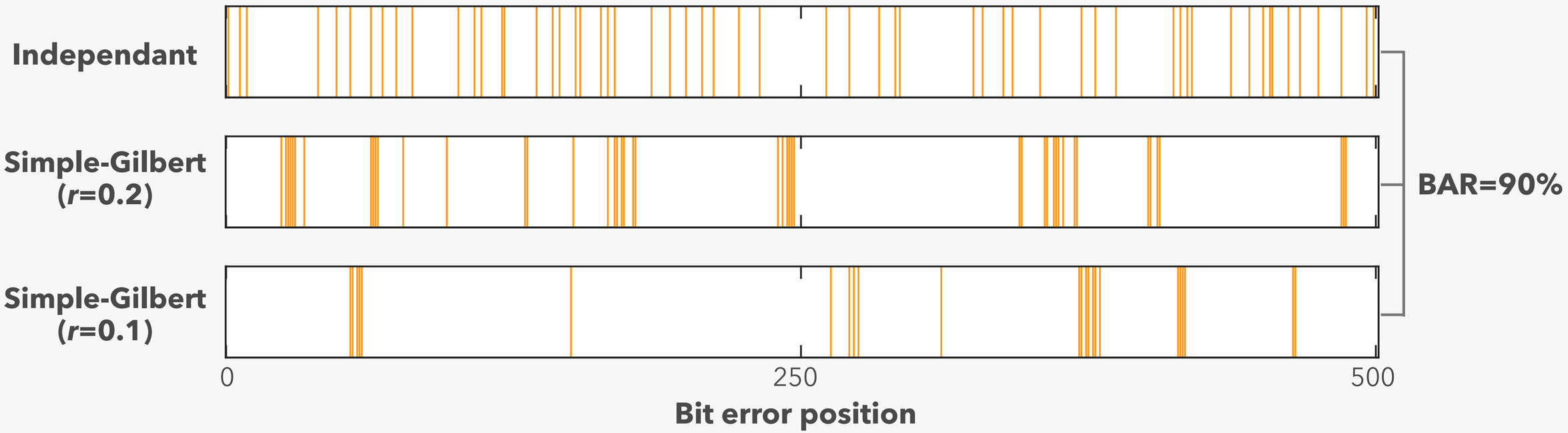
KEY RECONCILIATION SCHEME (CS–BASED)

	Device *A* (Host)		Device *B* (Client)	
→ 1)	$K_A = B_A$		$K_B = B_B$	
→ 2)	$C_A = \Phi(k \times M) \cdot K_A$		$C_B = \Phi(k \times M) \cdot K_B$	
→ 3)		C_A, SHA256(K_A) →	$\Delta C = C_A - C_B$	←
4)			ΔB = L1.min (ΔC)	←
5)			$K_B = B_B \oplus \Delta B$	←
6)			HASH(K_A) = HASH(K_B)	

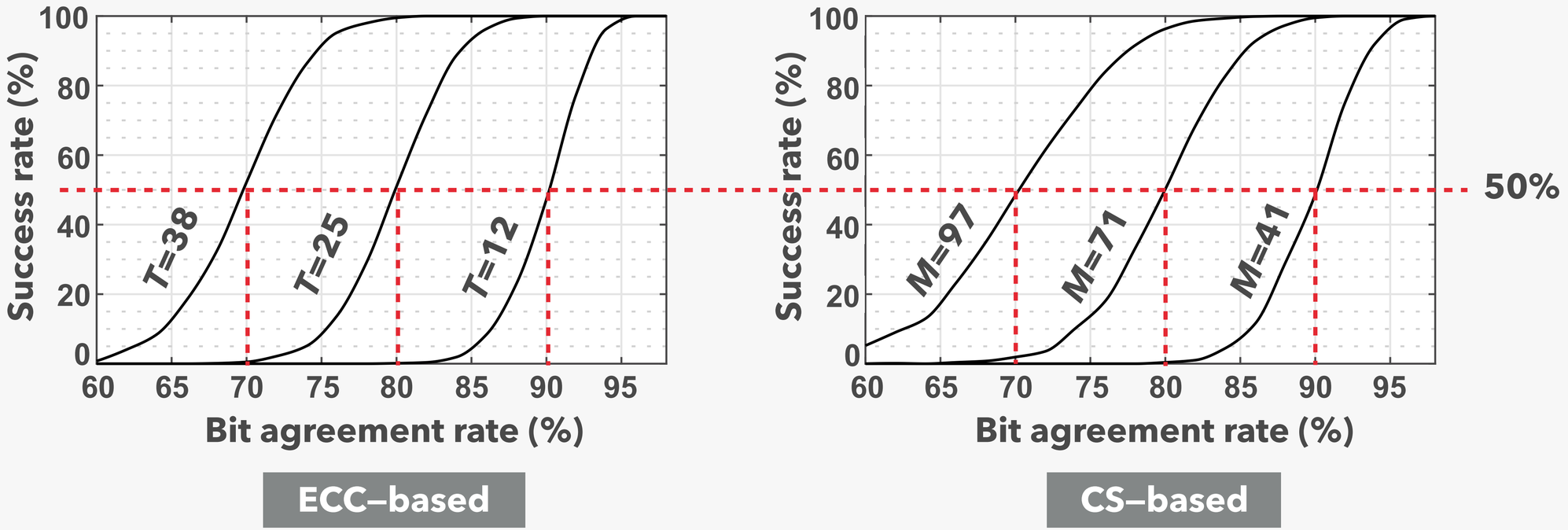
- Φ = sensing matrix of dimension $k \times M$
- C = compressed Key
- M, the number of non-compressed bits in Φ, is the parameter

Characterization and Analysis

- **Reconciliation performance, resulting entropy, and computation with *k* = 128 bits**
- **100,000 bits on two bit error models with BAR ranging from 60% – 97%:**
  - **Independent:** equal chance of error in each bit position
  - **Burst:** Simple-Gilbert model with *r* (probability of transitioning from bad to good state)



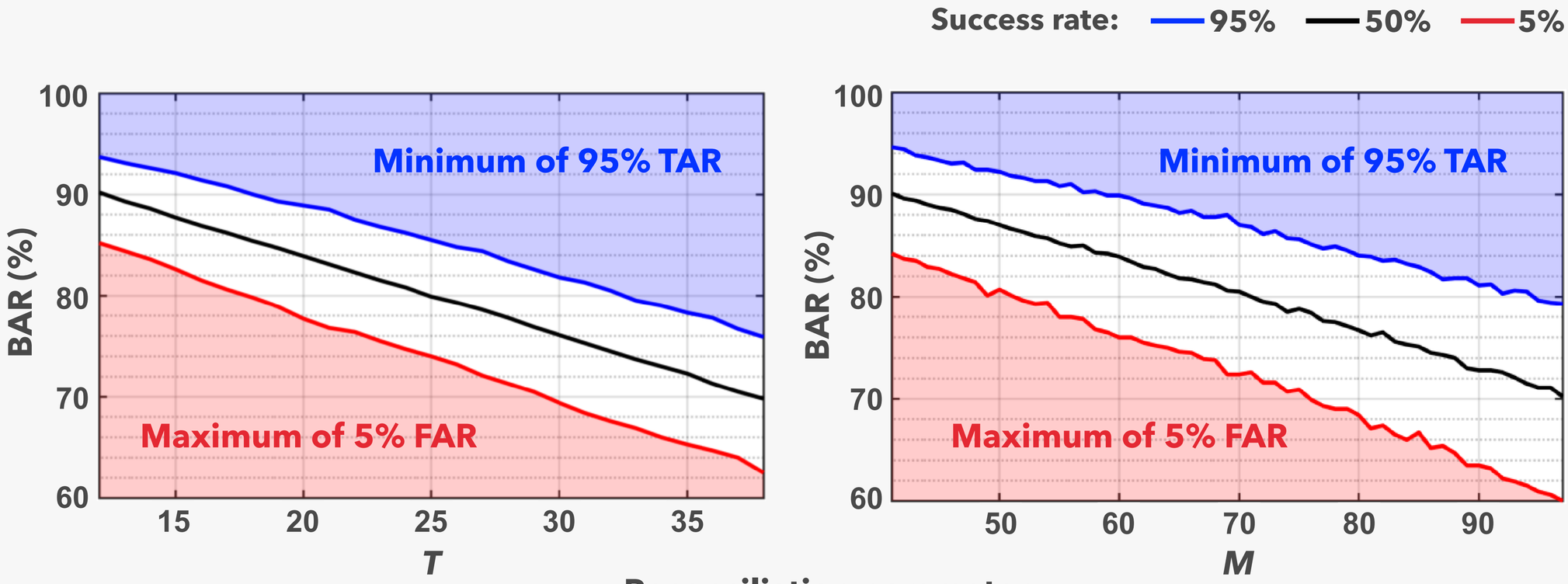
Reconciliation success rate



- As parameters *T* and *M* increases, it can correct more number of errors

_T_ of 12–38 exhibit equivalent success rate compared to _M_ of 41–97

RECONCILIATION SUCCESS RATE



ECC–based

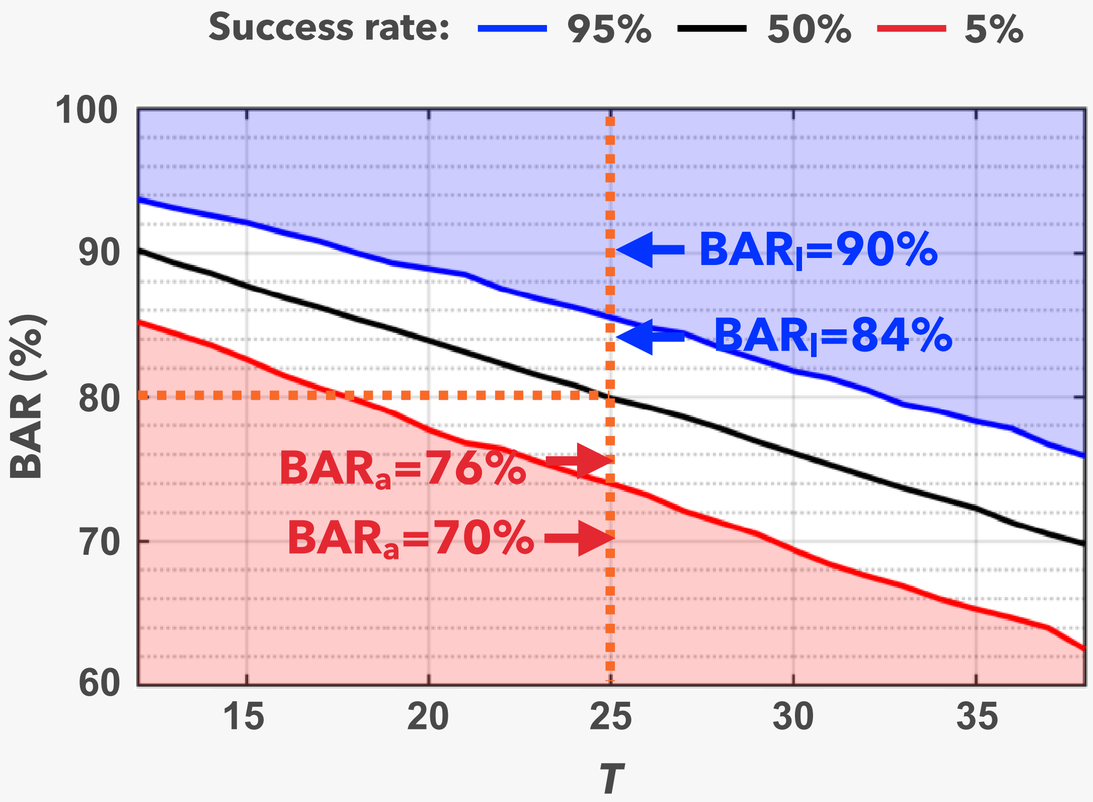
Shaded area=71.3%

CS–based

Shaded area=63.7%

ECC–based scheme shows better reconciliation performance

PARAMETER SELECTION



5% EER achievable

$BAR_l = 90\%$ $BAR_a = 70\%$

$mean(BAR_l, BAR_a) = 80\%$

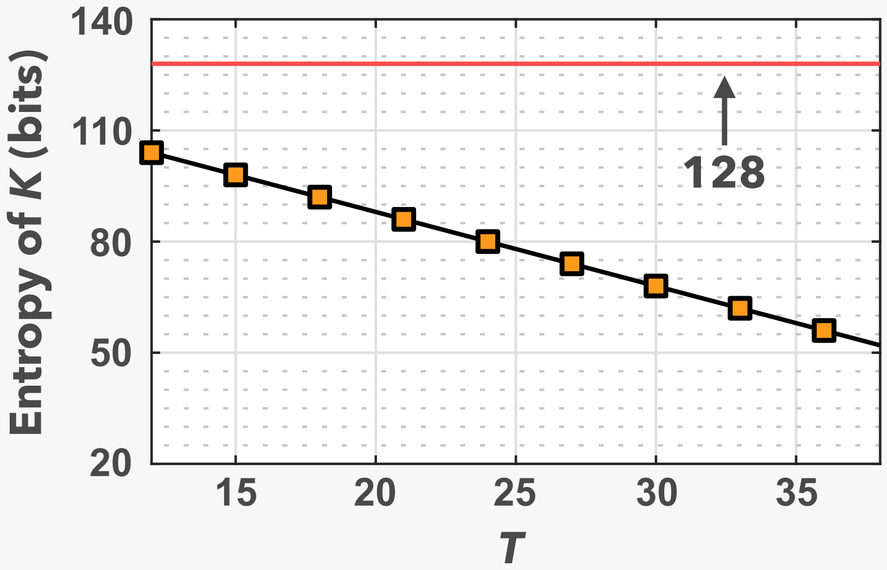
5% EER not achievable

$BAR_l = 84\%$ $BAR_a = 76\%$

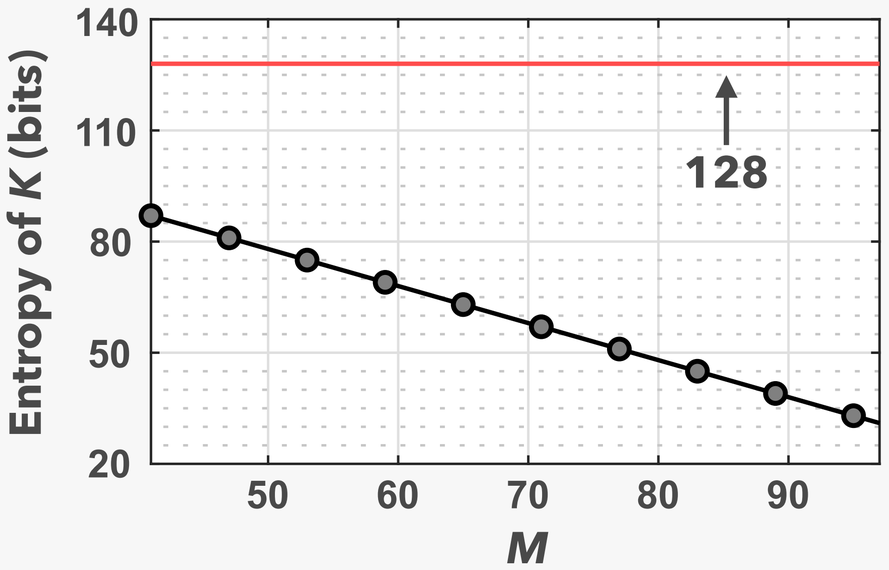
$mean(BAR_l, BAR_a) = 80\%$

- Take the mean of BAR_l and BAR_a
- Find *T* that corresponds to the mean BAR on 50% success rate line
- If BAR_l and BAR_a is both within shaded region (red and blue), 5% EER can be met

Retained entropy



ECC–based

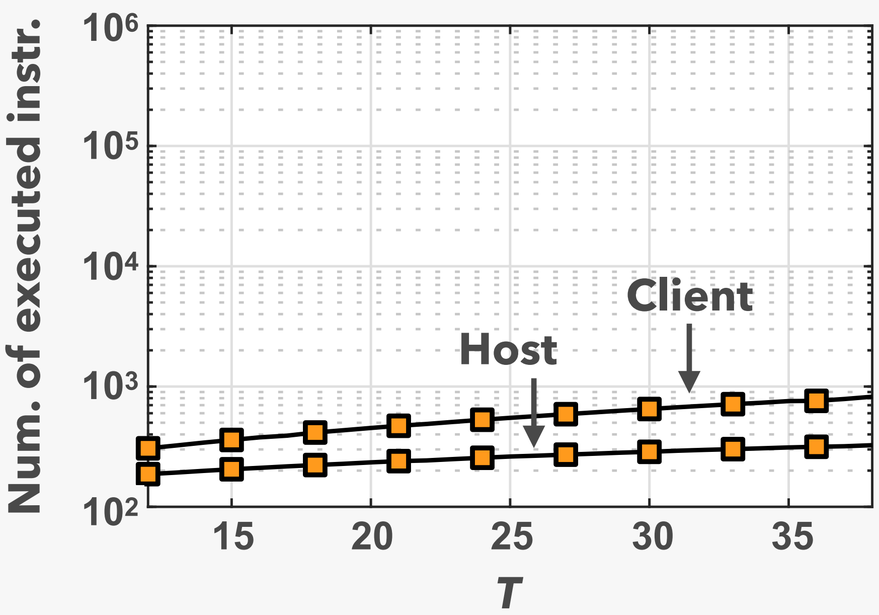


CS–based

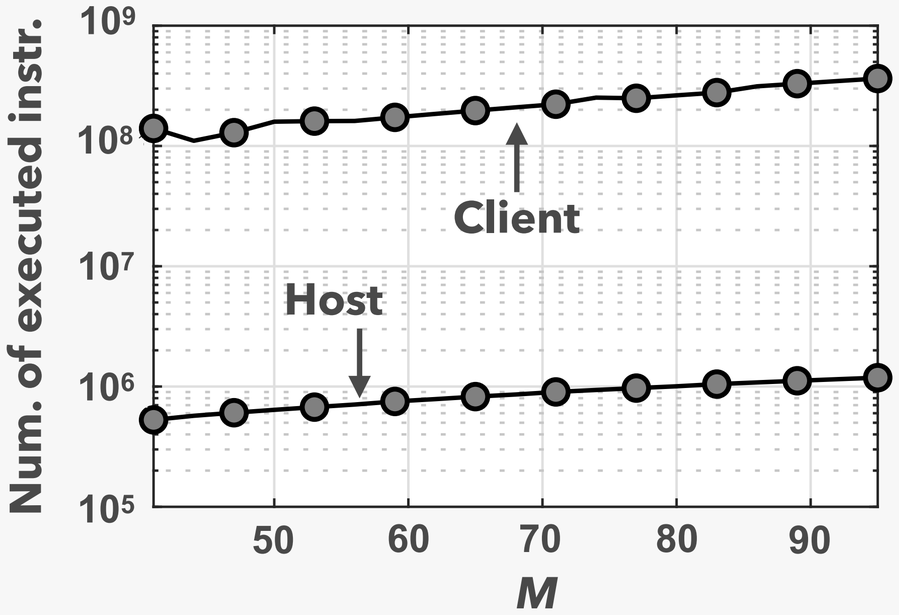
- Strong attack model where the adversary has access to pre-distributed information to reconcile final key

The final key of ECC–based scheme retains more entropy

COMPUTATION



ECC–based



CS–based

- Implemented in C and executed on Raspberry Pi 4 (ARM Cortex-A 1.4 Ghz)
- ECC–based: under 3.2 ms, CS–based: under 100 ms

ECC–based scheme is computationally lighter

Discussion

- **Selecting the reconciliation scheme**
  - ECC–based scheme outperforms the CS–based one in terms of reconciliation rate, retained entropy and computation
- **Framework implementation considering three bit error models with up to 10% target EER validation**
  - ECC–based: <2.5 kB of storage
  - CS–based: ~5 kB of storage
- **Our framework can**
  - Compare the performance of past and future ZIPA works
  - Provide a guideline for existing ZIPA developers
  - Be implemented on existing ZIPA works

Conclusion

- **We proposed a novel framework to determine a proper reconciliation parameter given user-defined authentication range**
  - Efficient and effective key reconciliation
  - Balanced security and usability
- **We analyzed the most commonly used schemes in terms of**
  - Reconciliation performance
  - Retained entropy
  - Computation
- **Help promote and explore systematic ZIPA pipeline**

Zero-interaction Pairing and Authentication (ZIPA)

- **Observing common contextual information means:**
  - Devices are in same place at same time
  - Devices belong to same user
- **Example of contextual informations include:**

RSSI

Audio

Luminosity

Image

Eliminates human-involvement during authentication

Motivation and Contribution

- **Existence of multiple key reconciliation scheme**
  - Error correcting code (ECC)–based
  - Compressed sensing (CS)–based
- **We need better understanding of current reconciliation schemes**
  - Error correcting performance
  - Computation
  - Entropy loss
- **Propose framework for ZIPA developers and existing ZIPA schemes to dynamically adjust authentication range and obtain proper parameter**
  - Usually most works just let the users determine the proper parameter