Enhancing NAC-ABE

Access Control for mHealth Applications and Beyond

Saurab Dulal[1] · Suravi Regmi[1] · Tianyuan Yu[2] · Adam Thieme[1] · Lan Wang[1] · Lixia Zhang[2]

[1] University of Memphis · [2] UCLA

Presented by

Suravi Regmi

Introduction

Why Wearable Data Sharing Is Hard

- Wearables continuously stream physiological data (heart rate, blood glucose, ECG, activity)
- Multiple stakeholders need access: doctors, trainers, researchers, caregivers
- Each stakeholder requires different visibility into the same data stream
- HIPAA and GDPR require this access to be fine grained and participant controlled

This calls for access control that is:

- Fine grained over data attributes (data stream, time, location, activity)
- Efficient at high frequency
- Flexible as policies evolve

Motivating Scenario: Alice

Alice's phone and wearables produce continuous streams of health data.

She wants to share with different parties under different rules.

Doctor

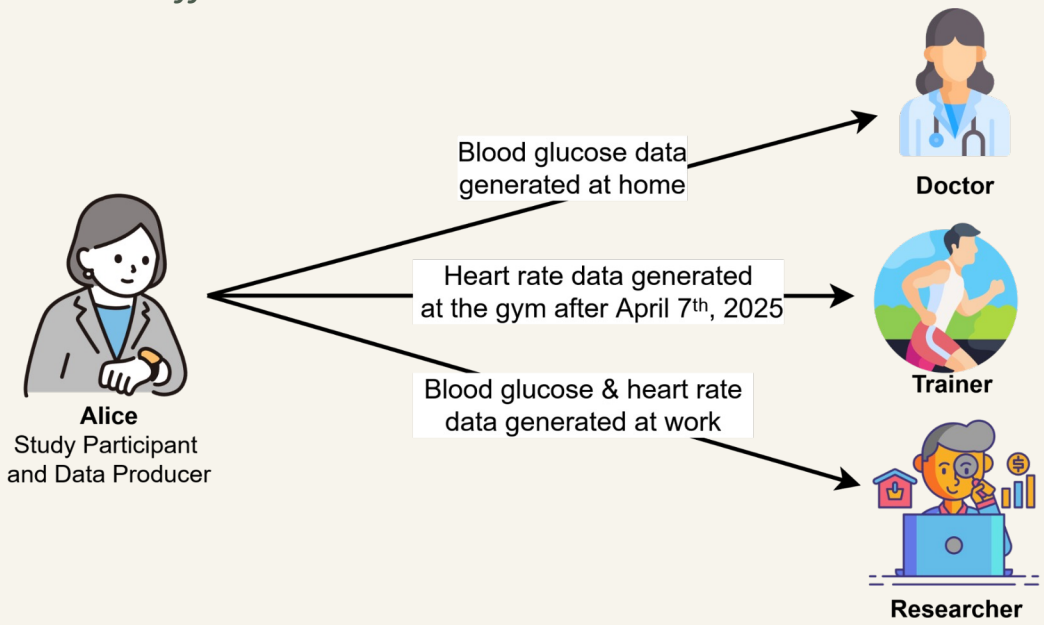
Blood glucose at home

Trainer

Heart rate at the gym, after April 7

Researcher

Both streams at work



Rules change: a nutritionist might be added later, requesting blood glucose access.

Requirements

mGuard is a secure, Real-Time mHealth Data Distribution application built on NDN and NAC-ABE.

For mGuard to support scenarios like Alice's, NAC-ABE must provide:

1. **Fine grained, contextual** access control over data stream, time, location, activity
2. **Efficiency** for high frequency streams (up to 120 Hz)
3. **Flexibility** for policy changes without re encrypting past data

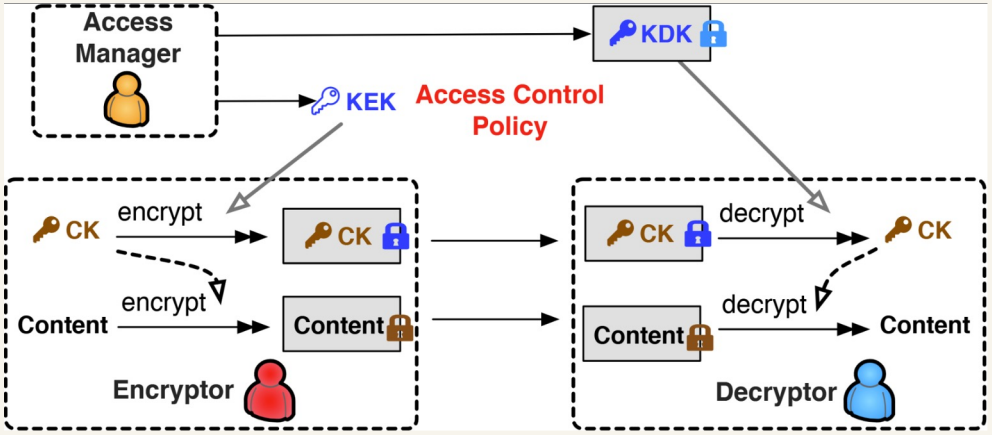
These three requirements drive every enhancement that follows.

Background: NAC

Name based Access Control enforces confidentiality in NDN by encrypting content using named keys.

How it works:

1. Producer encrypts content with a Content Key (CK)
2. CK is wrapped with a Key Encryption Key (KEK) from the Access Manager
3. Each consumer gets a Key Decryption Key (KDK), encrypted with their public key
4. Consumer decrypts the KDK with their private key, uses it to recover the CK, then decrypts the content



The scaling problem

For **m** consumers and **n** access granularities, NAC may require up to **m × n KDK packets**. Does not scale.

Background: NAC-ABE

Replaces NAC's public key based key distribution with **Attribute Based Encryption (ABE)**.

Attributes: descriptive strings that label data or users

"trainer" "gym" "heart-rate"

Policies: boolean expressions over attributes

e.g., ("trainer" AND "gym") or ("doctor" AND time > April 7)

NAC-ABE 2020 uses Ciphertext Policy ABE (CP-ABE):

- Data is encrypted with a policy that specifies who can decrypt it
- Each consumer's Decryption Key (DKEY) encodes that consumer's attributes
- Example: data encrypted with ("trainer" AND "gym"); the trainer's DKEY encodes "trainer" and "gym", satisfying the policy

Scaling win: for m authorized consumers, only **m DKEYs**, versus NAC's worst case of **m × n KDKs**.

But: the policy must be known and **fixed at encryption time**.

Limitations of NAC-ABE 2020

Four problems block real world mHealth deployment:

1. **Policies fixed at encryption time**

   Adding a new consumer means re encrypting all affected data.

2. **Naming scheme drifts from the original design**

   No support for key versioning or segmentation.

3. **One CK per data packet**

   360,000 CKs for one hour of data at 100 Hz.

4. **No trust schema validation**

   Signed but unverified, open to data injection.

Problem with CP-ABE

CP-ABE has two drawbacks in mHealth.

(a) Policy changes require re encryption.

If a nutritionist is later added to access Alice's blood glucose data, the access control policy must be updated and all affected CKs re encrypted and re published.

(b) Attributes can leak across policy clauses.

A researcher is authorized for:

`blood-glucose @ work`

`heart-rate @ home`

DKEY holds all four attributes:

{ blood-glucose, work, heart-rate, home }

Now blood-glucose at home arrives,

encrypted with policy:

("blood-glucose" AND "home")

Researcher has both "blood-glucose" and "home"

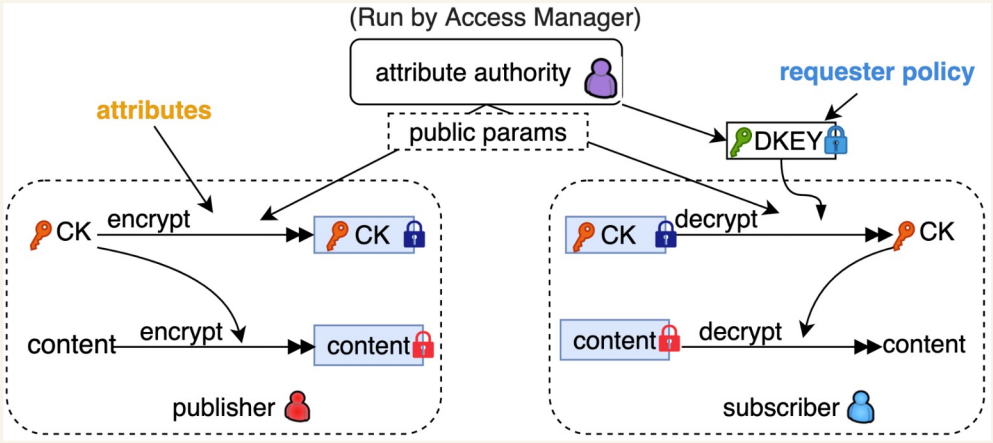
→ **wrongly gains access**

Attributes from different policy clauses combine unintentionally.

Fix 1: KP-ABE

In KP-ABE, the roles swap:

- Producers encrypt with data attributes
- DKEYs encode the consumer's access policy



Same researcher, now with KP-ABE:

Data attributes: `{ blood-glucose, home }`

Researcher's policy: `(blood-glucose AND work) OR (heart-rate AND home)`

Neither clause satisfied : **access denied**

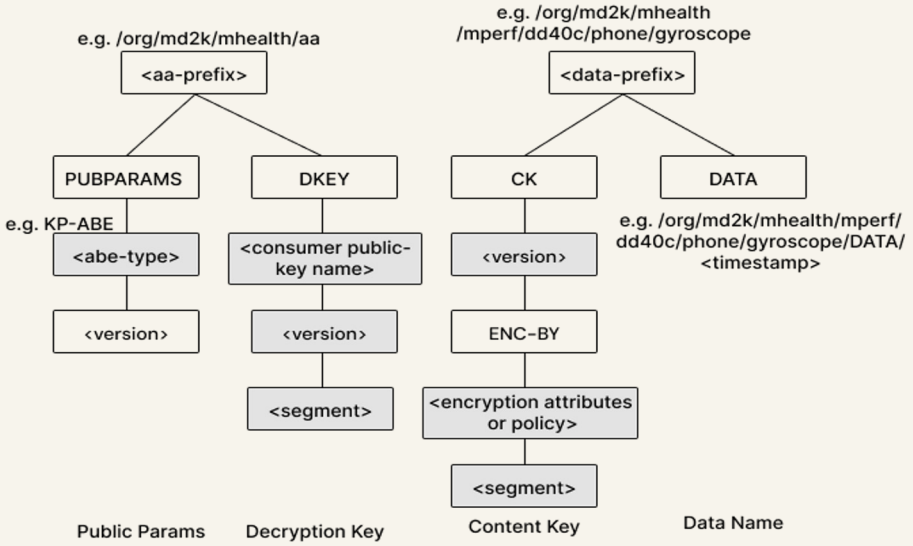
Adding a nutritionist: just issue a new DKEY. No re encryption.

Fix 2: Updated Naming Scheme

Realign with the original NAC-ABE design, extend for KP-ABE and segmentation.

Three key changes:

1. **`<abe-type>`** in public parameter names: supports both CP-ABE and KP-ABE.
2. DKEYs identified by **consumer's public key name** rather than identity: accommodates key rotation.
3. **Version and segment components** added to DKEY names: key evolution and segmentation.


e.g. /org/md2k/mhealth/aa
<aa-prefix>
PUBPARAMS
DKEY
e.g. KP-ABE
<abe-type>
<version>
<consumer public-key name>
<version>

Public Params
Decryption Key
e.g. /org/md2k/mhealth /mperf/dd40c/phone/gyroscope
<data-prefix>
CK
DATA
<version>
ENC-BY
<encryption attributes or policy>

Content Key
e.g. /org/md2k/mhealth/mperf/ dd40c/phone/gyroscope/DATA/ <timestamp>
Data Name

Why segmentation matters

NAC-ABE relies on the OpenABE library, which encodes timestamps as 32 bit UNIX integers and expands them into O(n) string attributes. With time based comparisons, CKs and DKEYs can far exceed the Maximum Segment Size.

Fix 3: CK Reuse and Granularity

Previously, one CK was generated per data packet. At 100 Hz, that is **360,000 CKs per hour**.

Solution: reuse CKs over a configurable time granularity.

Encryptor caches CKs per attribute set, reuses for matching packets.

Decryptor caches decrypted CKs to skip repeated fetch and decrypt.

Granularity must match the policy boundary.

Trainer authorized from April 7 onward:

- Monthly CK leaks earlier data
- Daily CK aligns with policy
- Hourly or per second tightens further, more overhead

Applications choose the granularity: fine for sensitive data (glucose), coarse for less sensitive (step counts).

Fix 4: Trust Schema Validation

NAC-ABE 2020 signs packets but never verifies them against a schema, leaving the system open to unauthorized data injection.

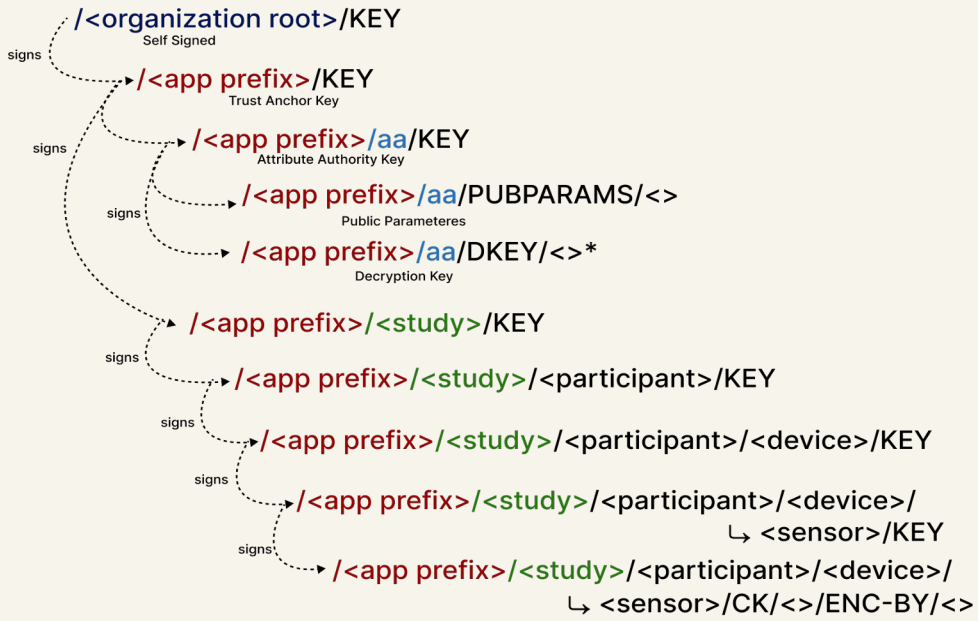
Solution: a validator interface checks every fetched packet against a trust schema.

Trust hierarchy

A chain of signing keys from the trust anchor down to each sensor, so every CK, DKEY, and public parameter traces back to a known root.

Containment

A compromised sensor only affects the CKs for its own data stream. Other data streams stay secure.



Evaluation Setup

Goal

Evaluate the improved NAC-ABE on:

- High frequency data as a realistic workload
- Different policies and CKs across multiple consumers
- Cryptographic cost characterization

Setup

NDN testbed multi consumer policy experiments

Mini NDN emulator cryptographic cost and CK experiments

Cerebral Cortex sensor streams at 1 Hz and 100 Hz

CK granularity hourly, minute, or second

VM running Ubuntu 20.04

Data Encryption Cost

Symmetric encryption and decryption stay sub-millisecond across all packet sizes.

At a 5.4 KB packet: **0.045 ms** to encrypt, **0.12 ms** to decrypt.

Data Size (B)	Enc. Size (B)	Encrypt (ms)	Decrypt (ms)
82	379	0.006	0.011
902	1,474	0.022	0.029
1,805	2,690	0.024	0.044
3,605	5,081	0.035	0.081
5,409	**7,490**	**0.045**	**0.120**

Data encryption / decryption time vs. data size.

CK Cost Grows with Attributes

CK generation is fast and flat (~1.8 ms regardless of attributes).

But adding a time attribute blows up CK encryption and decryption.

Attributes	Generation (ms)	Encrypt (ms)	Decrypt (ms)	Size (B)
S	1.81	2.92	10.01	841
S, L	1.88	3.22	10.20	981
S, T	**1.89**	**15.43**	**23.64**	**3,260**
S, T, L	1.89	15.26	23.36	3,400
S, T, L, A	1.89	16.30	24.30	3,613

S = stream, T = time, L = location, A = activity

CK operation costs and size vs. attribute set.

OpenABE expands UNIX timestamps into O(n) string attributes.

The number of CK operations becomes the key performance factor.

CK Reuse

Previously: 100 Hz stream → 360,000 CKs per hour → **over 4 hours of CK computation.**

CK count depends only on granularity, not sampling frequency.

Granularity	CKs per hour	CK compute
Per second	3,600	146 s
Per minute	60	2.4 s
Per hour	**1**	**0.04 s**

For a 100 Hz stream: CK overhead drops from over 4 hours per hour of data to 0.04 seconds.

Combined with caching on both producer and consumer sides, each CK is generated, fetched, and decrypted only once per granularity window.

This is what enables mGuard to support real time encrypted data sharing across streams at varying frequencies.

Applicability Beyond mHealth

The enhancements apply wherever data has known attributes but evolving policies:

- **Smart buildings:** floor, zone, time across HVAC, occupancy, energy
- **Military sensor networks:** unit, mission, access tier
- **Vehicular NDN:** speed, location, diagnostics across insurance, traffic systems, emergency responders

Common pattern: data attributes are known at production, but who should access them evolves over time.

Discussion and Open Issues

Access revocation

- Time bounded DKEYs give eventual, not immediate revocation
- Data published within an active DKEY window stays accessible to a revoked consumer
- HIPAA mandates participants can withdraw consent at any time
- Directions: shorter DKEY windows, proxy re encryption, NDN native revocation

Resource constraints

- OpenABElibrary is no longer actively maintained, a real deployment risk
- Looking into RABE, a maintained Rust based alternative, going forward
- Mobile deployment adds key storage and battery concerns

Conclusion

We addressed four limitations in NAC-ABE 2020:

CP-ABE's policy at encryption restriction → **KP-ABE addition**

Naming scheme inconsistencies → **realigned and extended for versioning and segmentation**

Per packet CK overhead → **CK reuse, caching, and configurable granularity**

Missing trust schema validation → **validator interface across all fetched data**

Looking ahead: Revocation remains the open problem. mHealth needs consent aware access control where withdrawal applies to already published data, a governance challenge that motivates our next steps.

References

- Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, Named Data Networking, ACM SIGCOMM CCR, 2014
- Saurab Dulal, Nasir Ali, Adam R. Thieme, Tianyuan Yu, Sichen Liu, Suravi Regmi, Lixia Zhang, Lan Wang, Building a Secure mHealth Data Sharing Infrastructure over NDN, ACM ICN 2022
- Zhiyi Zhang, Yingdi Yu, Sanjeev Kaushik Ramani, Alexander Afanasyev, Lixia Zhang, NAC: Automating Access Control via Named Data, MILCOM 2018
- Yingdi Yu, Alexander Afanasyev, Lixia Zhang, Name-based Access Control, NDN Tech Report NDN-0034, 2015
- Yingdi Yu, Alexander Afanasyev, David Clark, KC Claffy, Van Jacobson, Lixia Zhang, Schematizing Trust in Named Data Networking, ACM ICN 2015
- Amit Sahai, Brent Waters, Fuzzy Identity-Based Encryption, EUROCRYPT 2005
- John Bethencourt, Amit Sahai, Brent Waters, Ciphertext-Policy Attribute-Based Encryption, IEEE S&P 2007
- Vipul Goyal, Omkant Pandey, Amit Sahai, Brent Waters, Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data, ACM CCS 2006
- Joseph A. Akinyele, Matthew W. Pagano, Matthew D. Green, Christoph U. Lehmann, Zachary N. Peterson, Aviel D. Rubin, Securing Electronic Medical Records Using Attribute-Based Encryption on Mobile Devices, SPSM 2011
- Matthew Green, Susan Hohenberger, Brent Waters, Outsourcing the Decryption of ABE Ciphertexts, USENIX Security 2011
- Bing Li, Dijiang Huang, Zhijie Wang, Yan Zhu, Attribute-Based Access Control for ICN Naming Scheme, IEEE TDSC, 2016
- OpenABE Library, Zeutro, github.com/zeutro/openabe
- NDN Project Team, Mini-NDN, github.com/named-data/mini-ndn

Thank you

Q&A