ICICV 2020 • Springer • Manipal University Jaipur

Attribute-Based Break-Glass Access Control

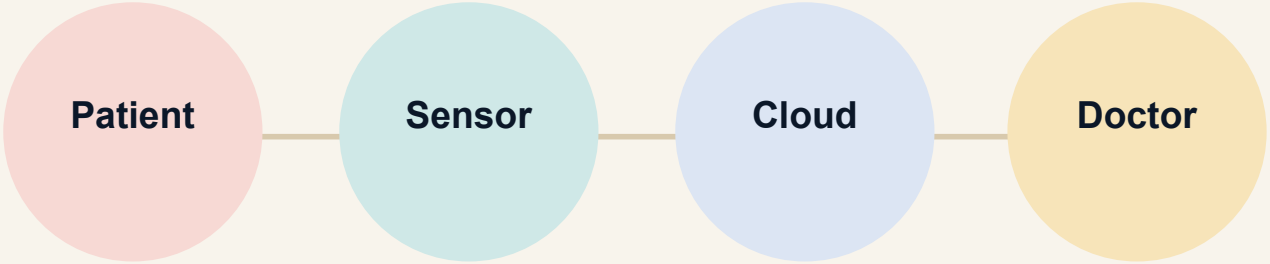
Framework for Medical Emergencies

Paper presented by Anubha Parashar
School of Computing & Information Technology

Why this work matters

IoT healthcare needs fast access and strong protection at the same time.

Wearable sensors and medical devices collect patient data continuously.
Cloud infrastructure enables doctors to review EHR data and support remote care.
Security failures in healthcare can expose sensitive data or delay treatment.



Connected care pipeline

Core tension

Routine access requires strict authentication; emergency care requires immediate access.

The emergency access challenge

Normal security policies are essential — but in emergencies, they can become a bottleneck.

Normal access

Attribute-based checks enforce who can decrypt and access sensitive data.

Emergency break-glass

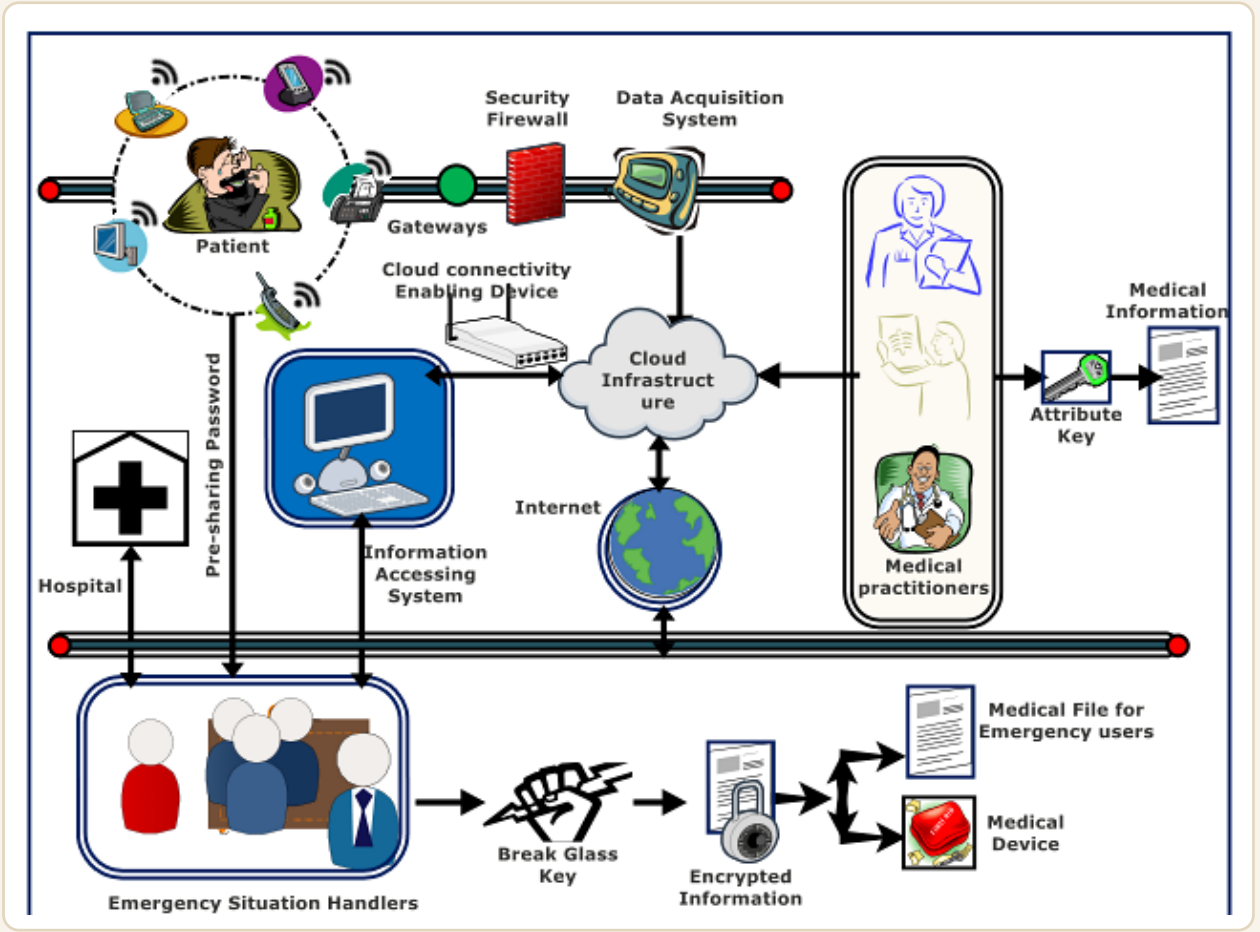
Emergency Situation Handlers need controlled bypass after verification and false-alarm screening.

Design goal: reduce delay without removing accountability.

Proposed framework architecture

Four cooperating actors create normal and emergency access pathways.

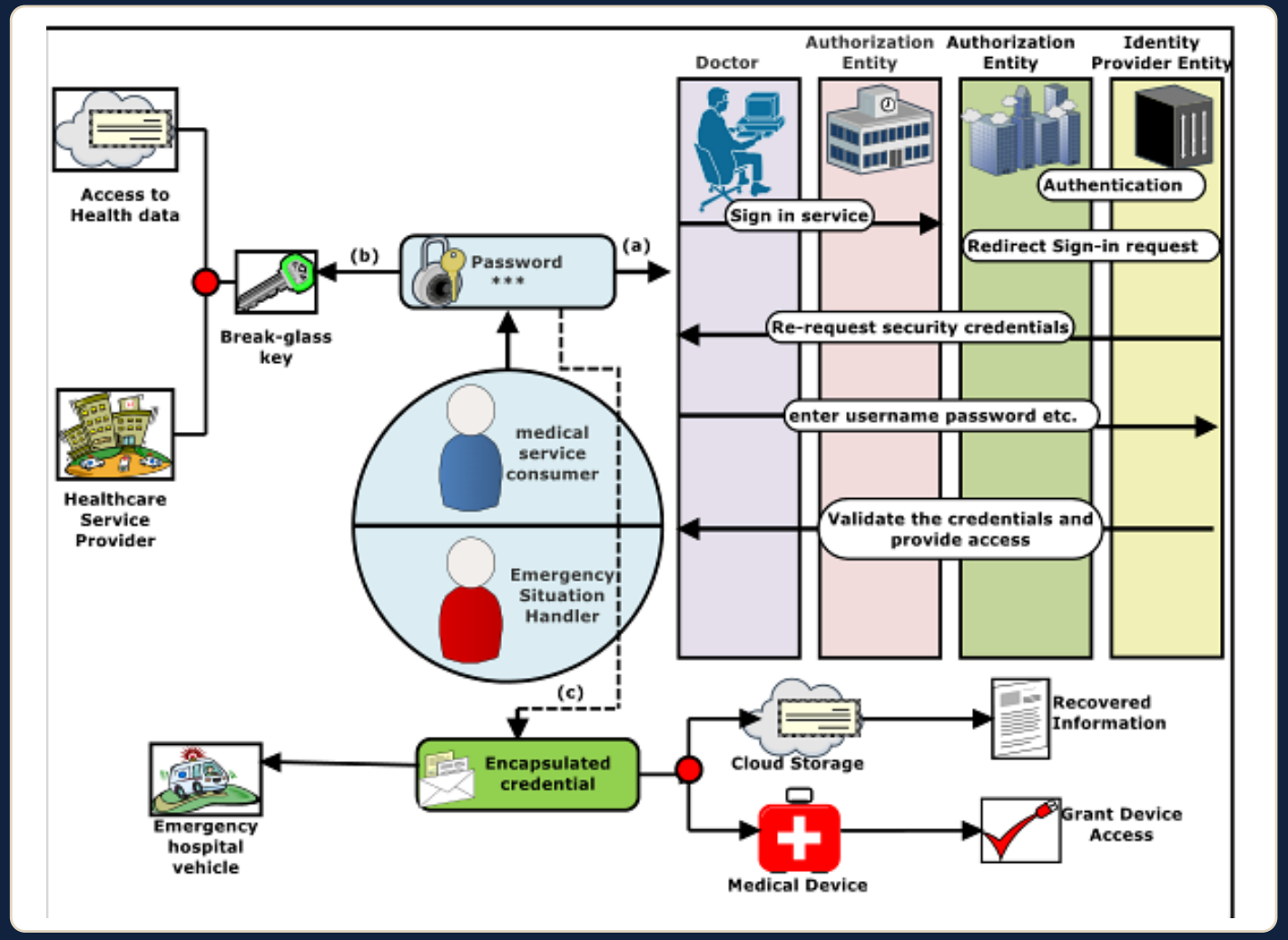
Medical Service Provider manages healthcare resources and devices.
Cloud Infrastructure stores encrypted medical information.
Medical Service Consumer receives care through IoT-enabled monitoring.
Emergency Situation Handler can access data/device only in verified emergency mode.



Key idea: normal attribute access + controlled break-glass access in one lightweight scheme.

Emergency key extraction workflow

The workflow verifies the handler, creates an Emergency Session Key, and grants bounded access.



1 Verify ESH credentials

2 Build emergency session key

3 Screen false alarms

4 Decrypt EHR + device access

Algorithm at a glance

BG.KeyExt keeps the emergency pathway controlled: authenticate first, then decrypt.

Fig. 3 demonstrates the procedural algorithmic representation of proposed scheme. Initially the patient shares the password (*pw*) for accessing his device and cloud data with the registered ESH. The ESH verification is done at the outer layer of the ***BG.KeyExt*** procedure. The peripheral function is attributed by ESH user credentials (cr1, cr2) such as email and resource access password etc. Once the credentials are verified at the CI database, then the legitimacy of the ESH user is identified. This step is to ensure the proposed security scheme cannot be used for misusing. In order to protect *pw*, ESH user utilizes the *pw* in peripheral function and generates encrypted password called emergency session key (ESK) as shown in Fig. 3. The attribute based access the users employs the different attributes such as *session key*, *UI, DoB* etc. The Authors invoked attribute based encryption-decryption scheme from [16] in order to utilize time effectively in emphasizing the design of break-glass access. Once the ESH gets the encrypted password EmSesKey, the same is attributed in inner function ***fun2*** along with ***pw*** as shown in Fig. 3. The ESH

```
Password-processed Break-Glass Key Extraction:BG.KeyExt
fun1 Peripheral _ Login.Service (cr1, cr2);

if (authorized!) in Peripheral_Login.service then
    return Error "Invalid ESH"
else
    return EmSesKey
    begin fun2  BG.KeyExt (Pw, EmSesKey)
    Input: Pre-shared patient password  (Pw), Emergency Session Key ( EmSesKey )
   Output: Decrypted Electronic Health Record (Dec.EHR)
    if  EmSesKey in False  Alarm  Entity then
```

Input — ESH credentials + patient password

↓

Validate — Reject invalid handler

↓

Check — Stop if false emergency

↓

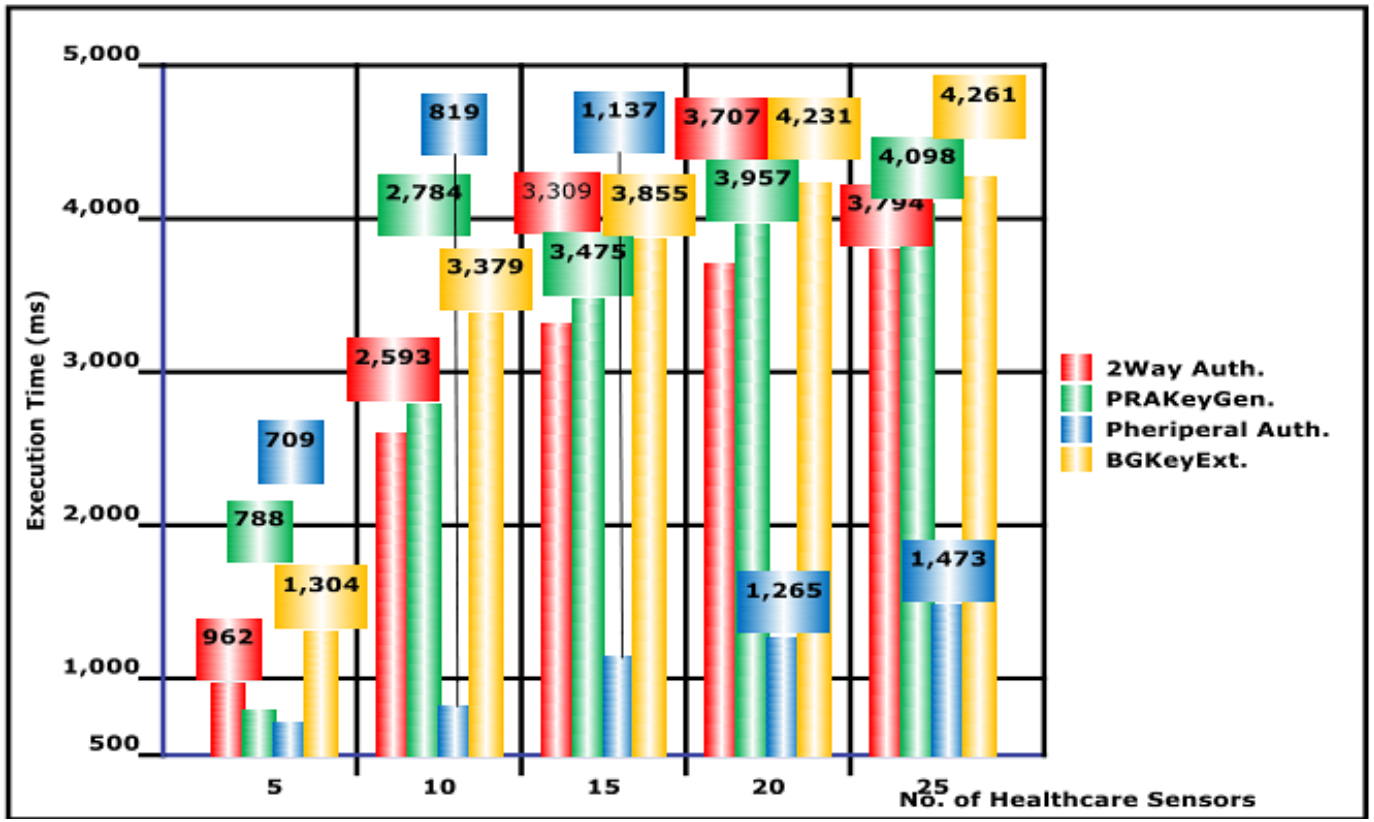
Authorize — Load role and ESH policies

↓

Output — Break-glass key and access rights

Result analysis: computational cost

Execution-time comparison across increasing numbers of healthcare sensors.



Observed trend

Multilayer authentication grows slower under low resource counts but scales poorly as sensors increase.

Key efficiency point

The proposed scheme is presented as lightweight, with minimal calculations at device and storage levels.

Practical impact

Lower latency supports emergency handling in resource-constrained IoT healthcare settings.

Security contribution

The design balances availability, authorization, and confidentiality.

1

Verified handler

Credentials are checked before emergency access begins.

2

False-alarm guard

Emergency access is refused when the event is flagged as false.

3

Encrypted EHR

Model argues resistance under DBDH-based security assumptions.

Contribution: a lightweight access-control path that handles normal data decryption and verified emergency override.

Takeaways

1. **Medical emergencies need availability, not just confidentiality.**
2. **Break-glass access must be verified, policy-bound, and auditable.**
3. **The proposed framework targets fast, lightweight emergency access for IoT healthcare.**

Presented at ICICV-2020 • Manipal University Jaipur • Springer

Paper: “An Attribute Based Break-Glass Access Control Framework for Medical Emergencies”