Using Generic Bootstrapping Architecture with Constrained Devices

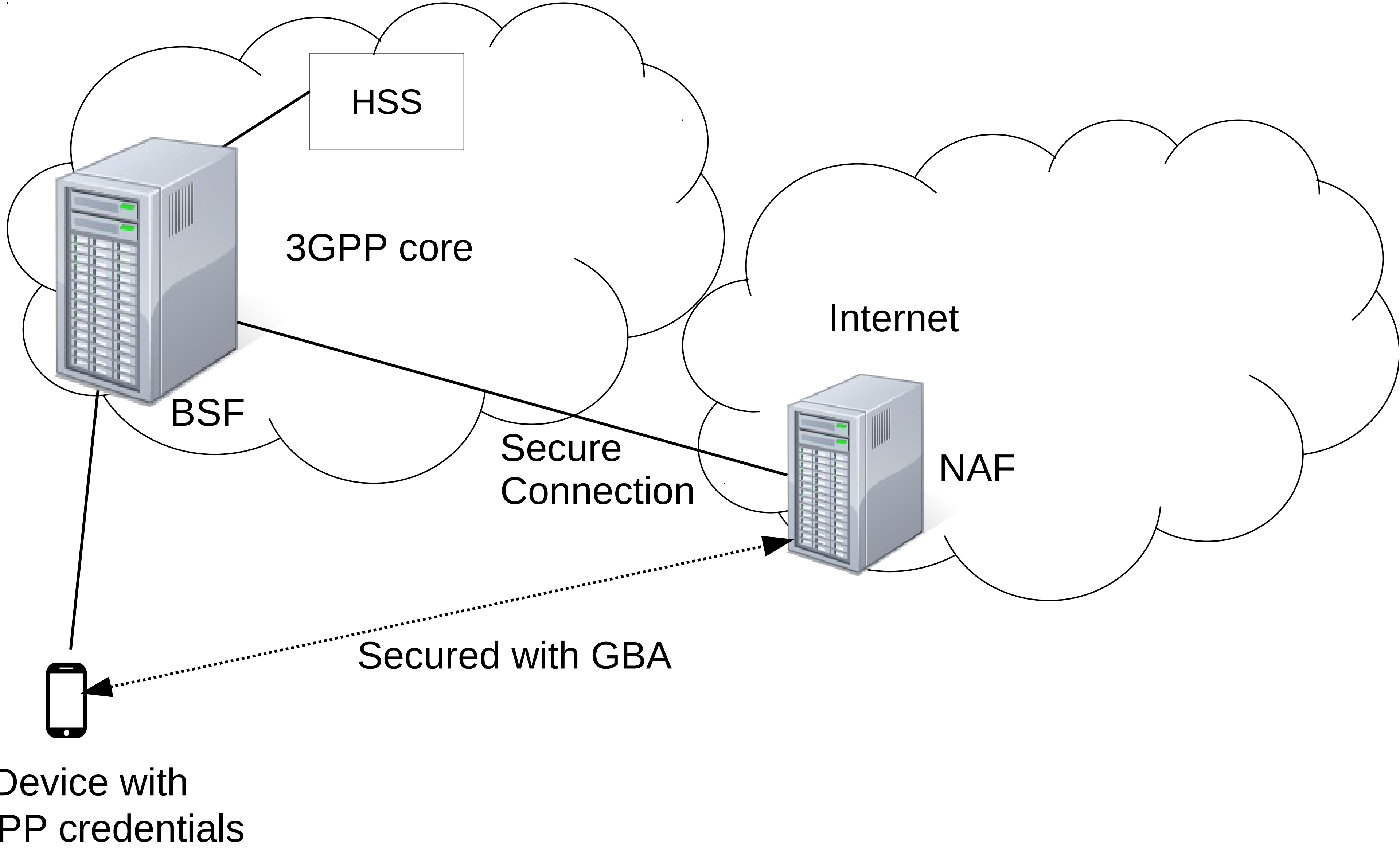
draft-sethi-gba-constrained-01

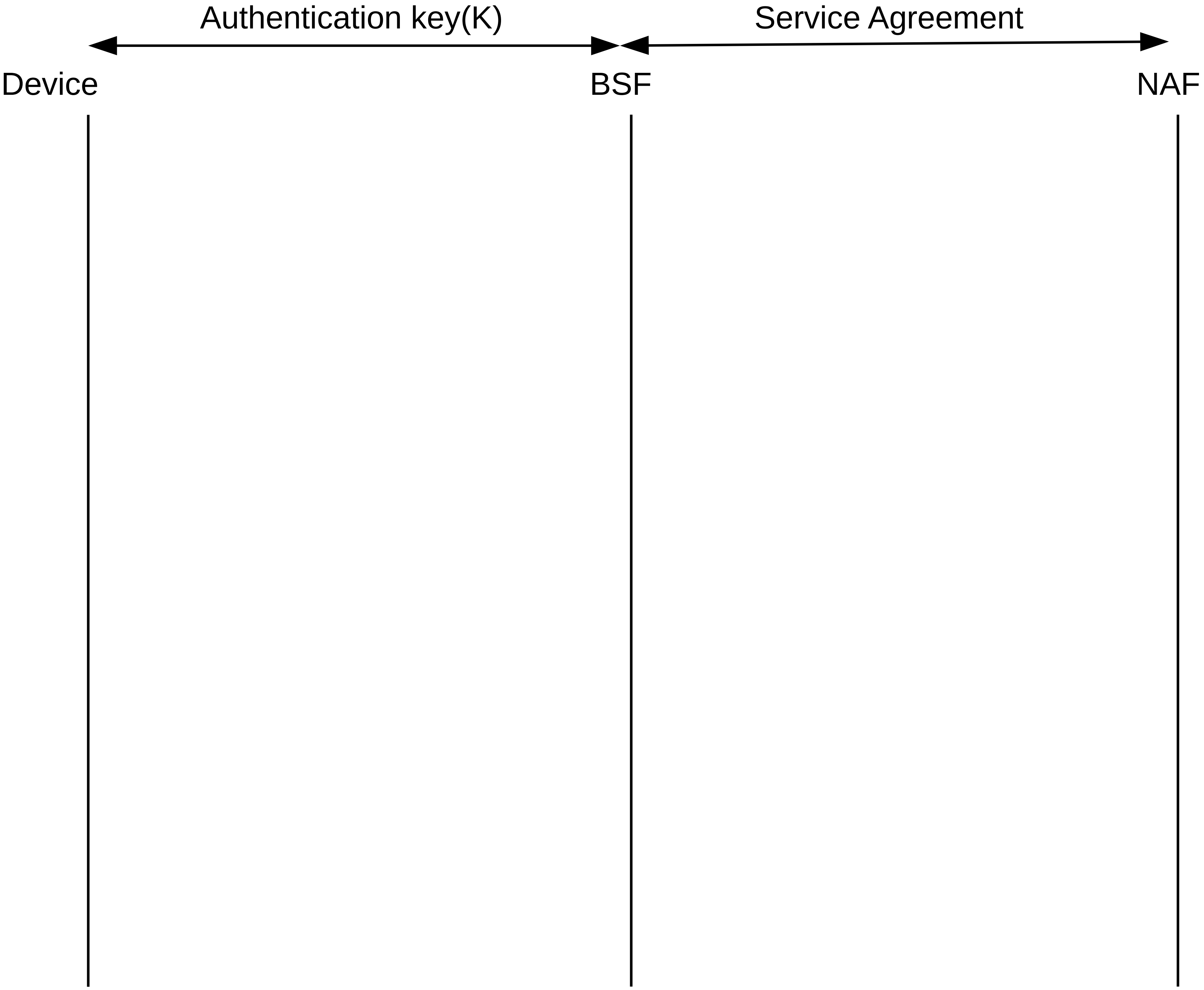
M. Sethi
V. Lehtovirta
P. Salmela
Ericsson

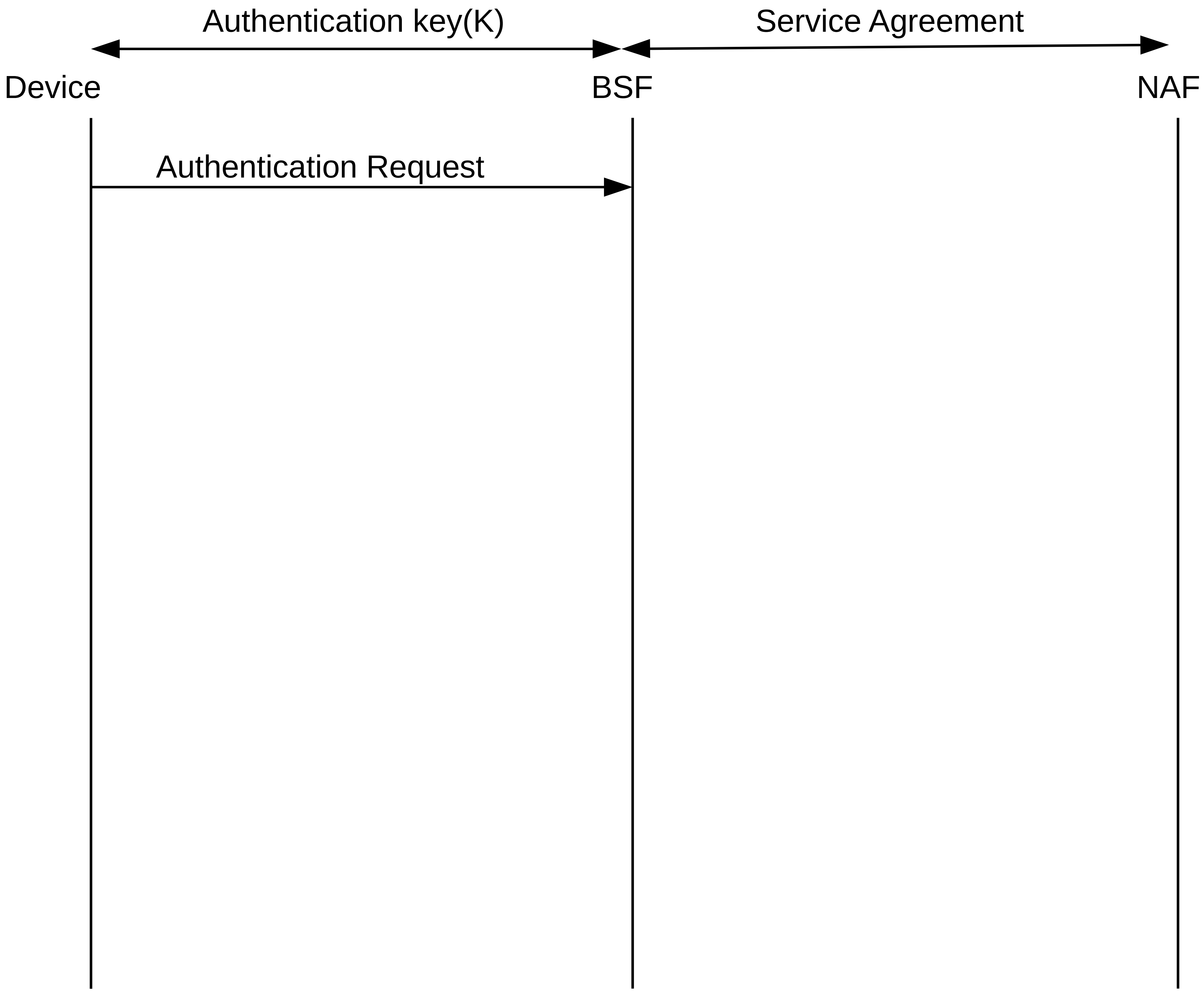
GBA basics

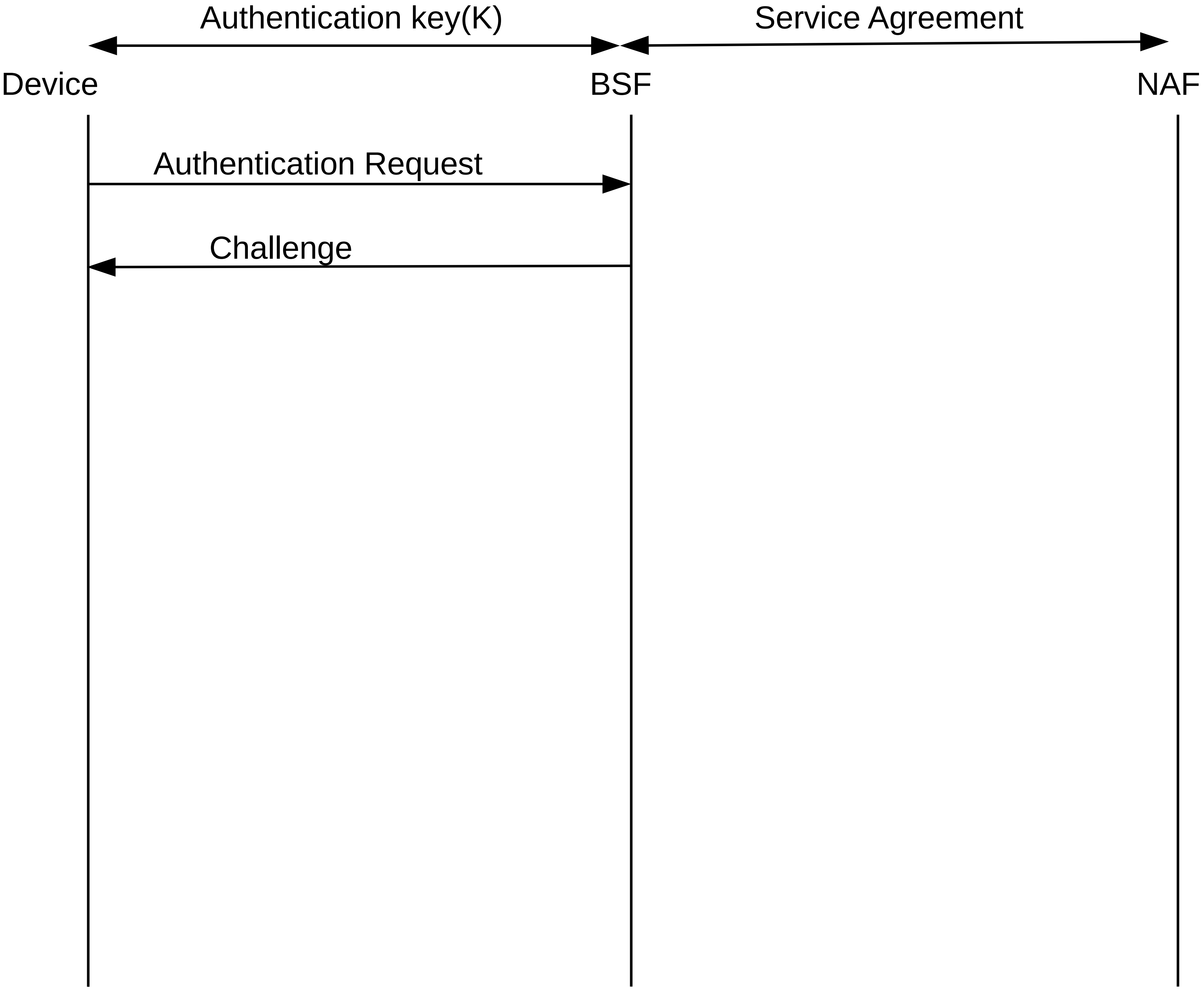
- Generic Bootstrapping Architecture
  - 3GPP TS 33.220
- Device **authenticates** to a service **using the SIM card**
    - Does not need to be done over 3GPP access, **any IP based connectivity**
- Kerberos like authentication system which is deployed

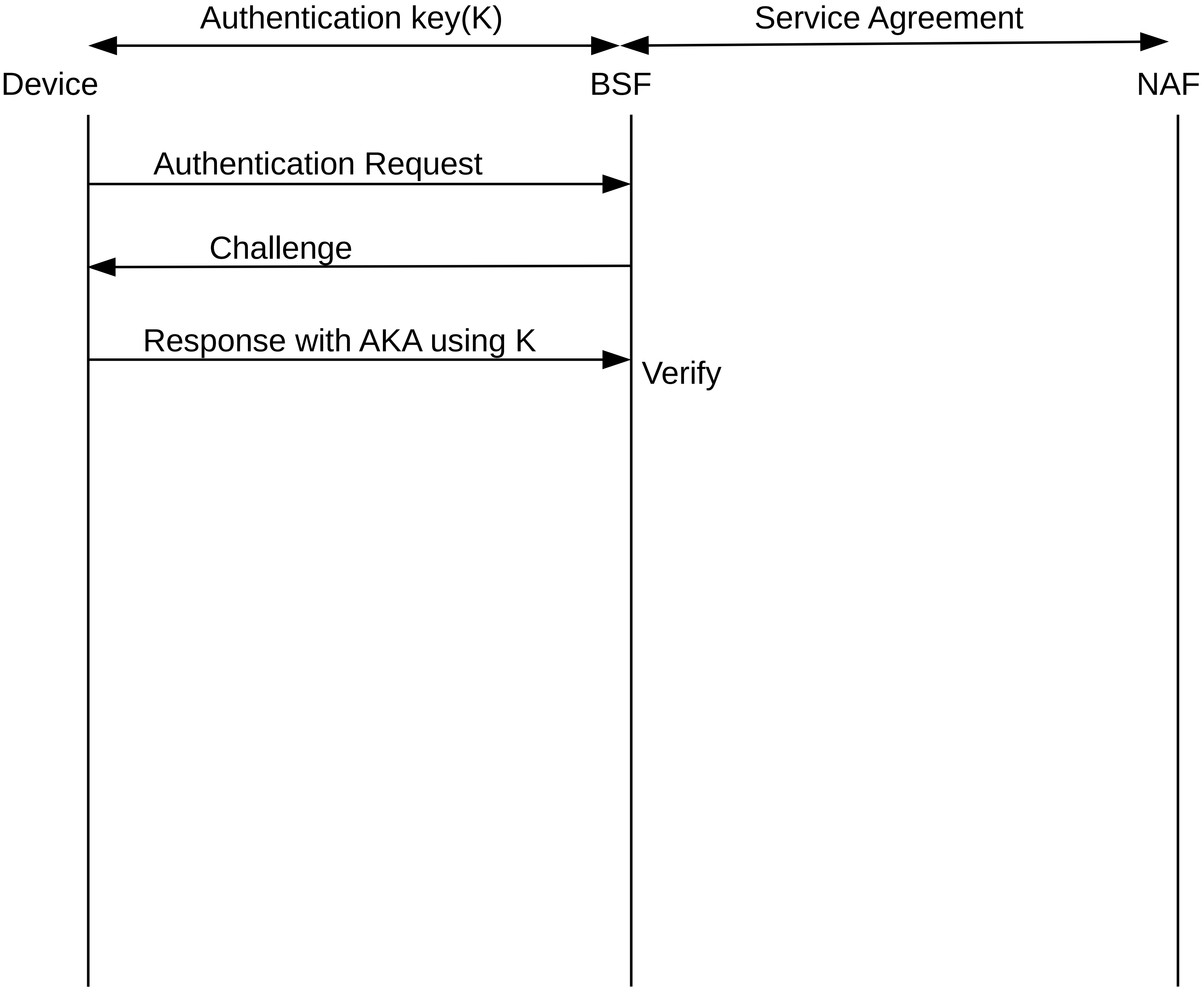
GBA basics

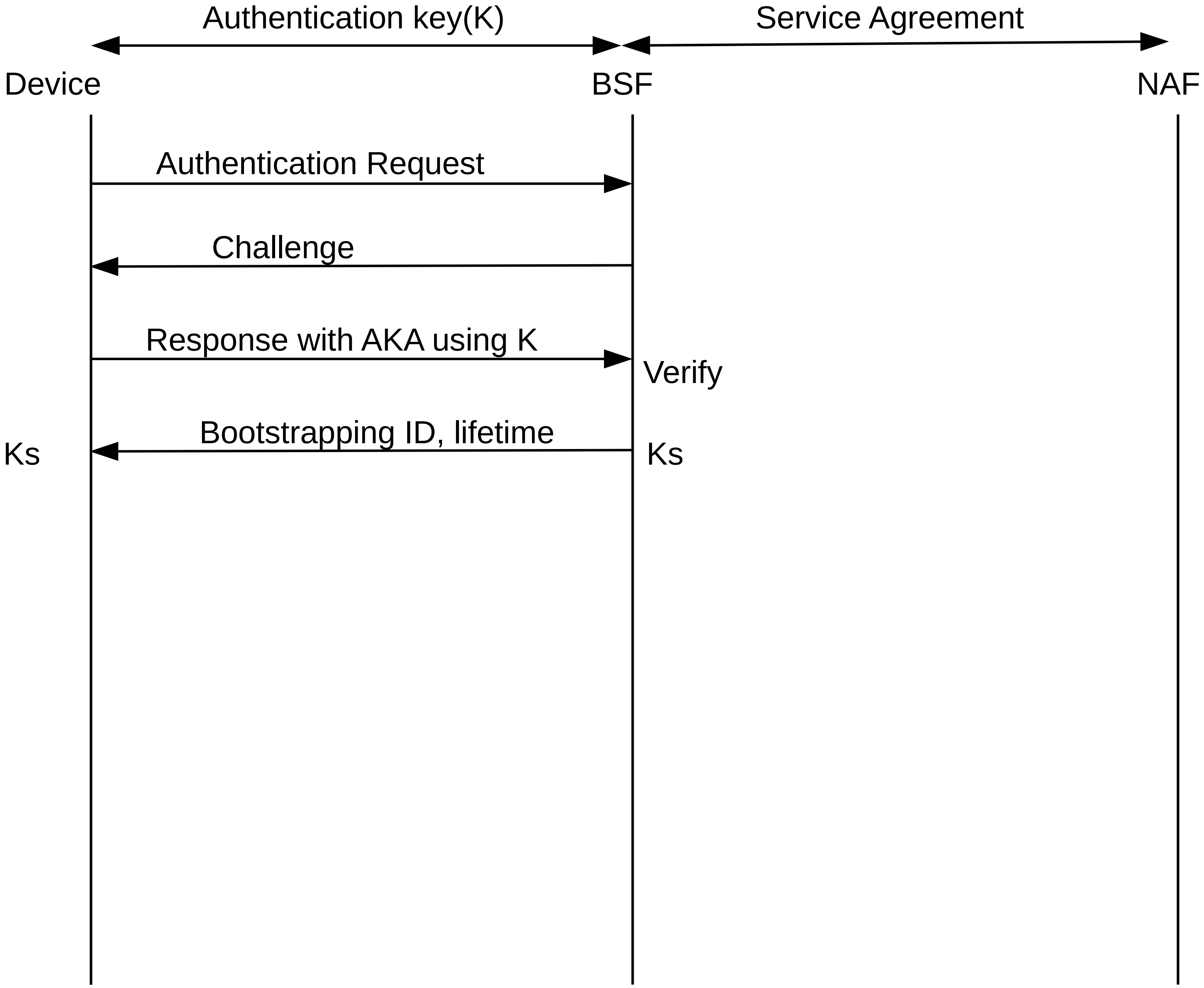

HSS
3GPP core
BSF
Internet
Secure
Connection
NAF
Secured with GBA
Device with
3GPP credentials

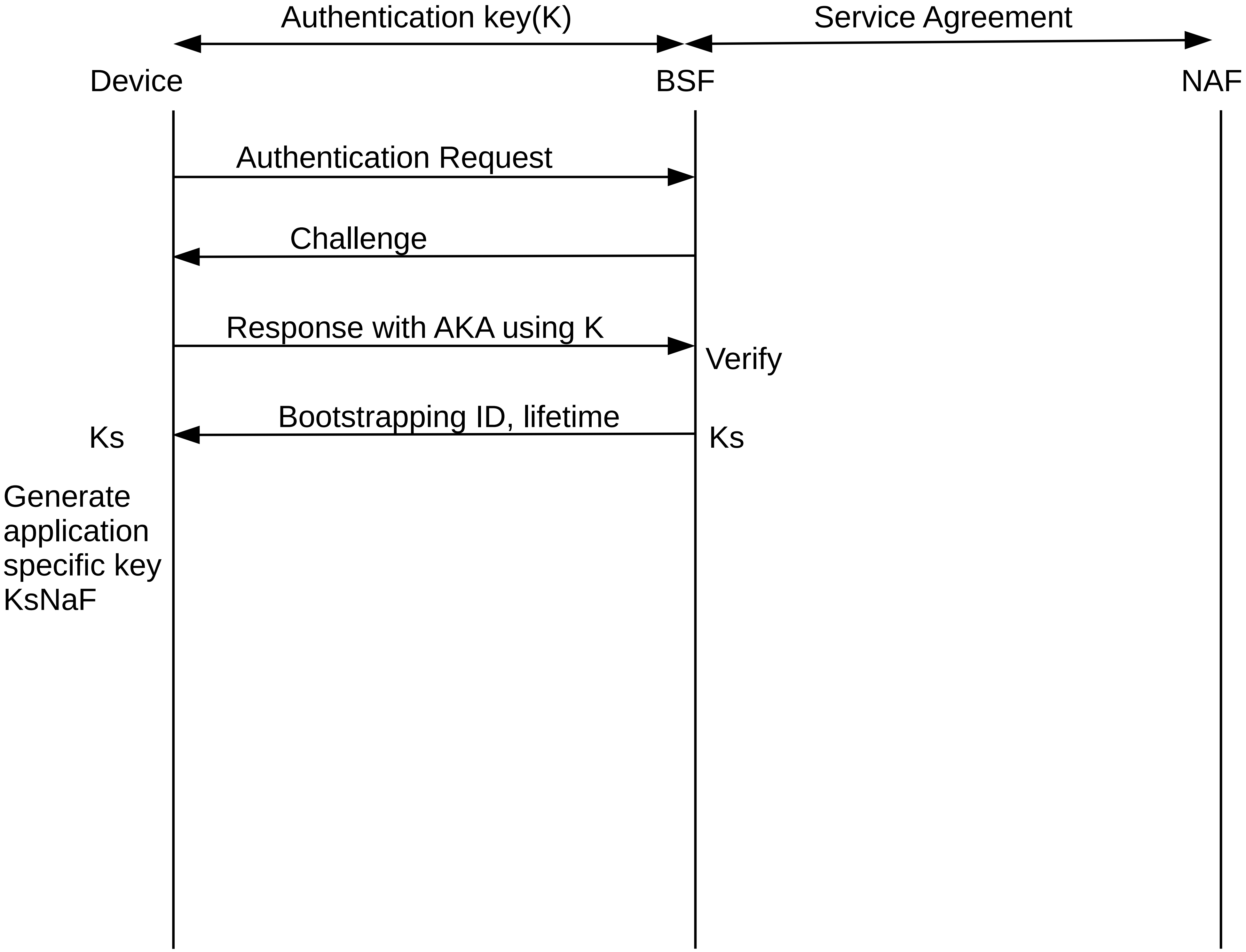

Authentication key(K)
Service Agreement
Device
BSF
NAF

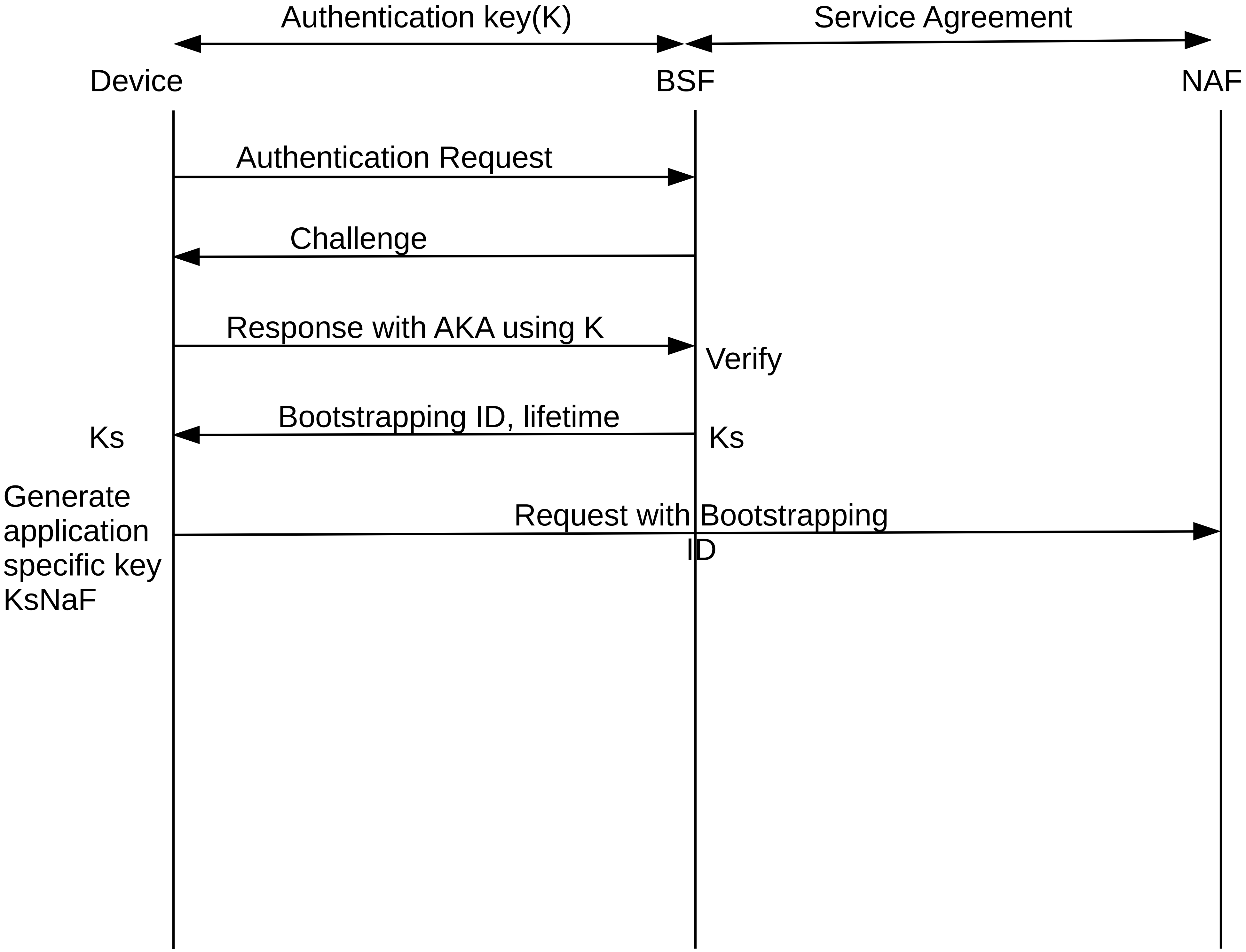

Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request

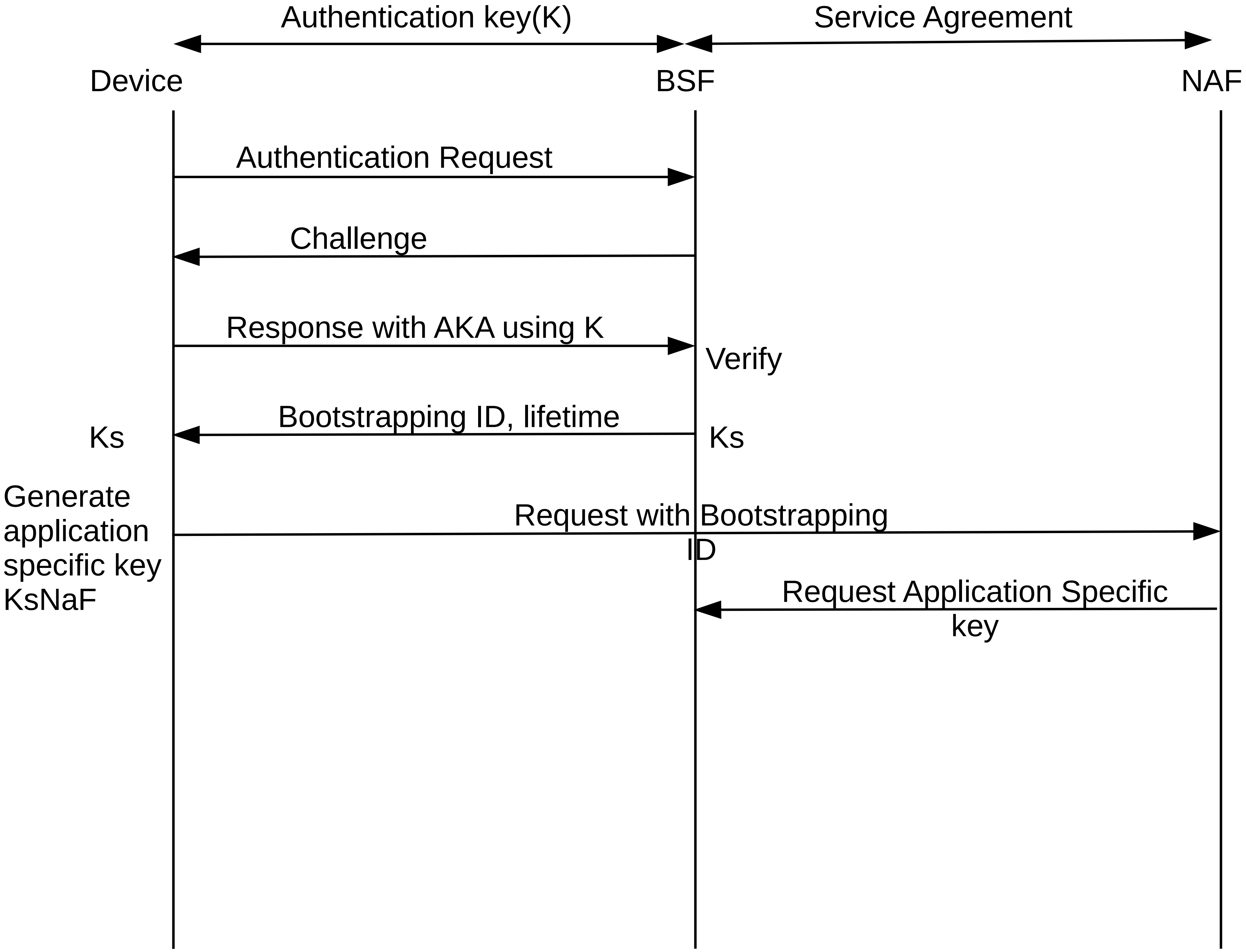

Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge

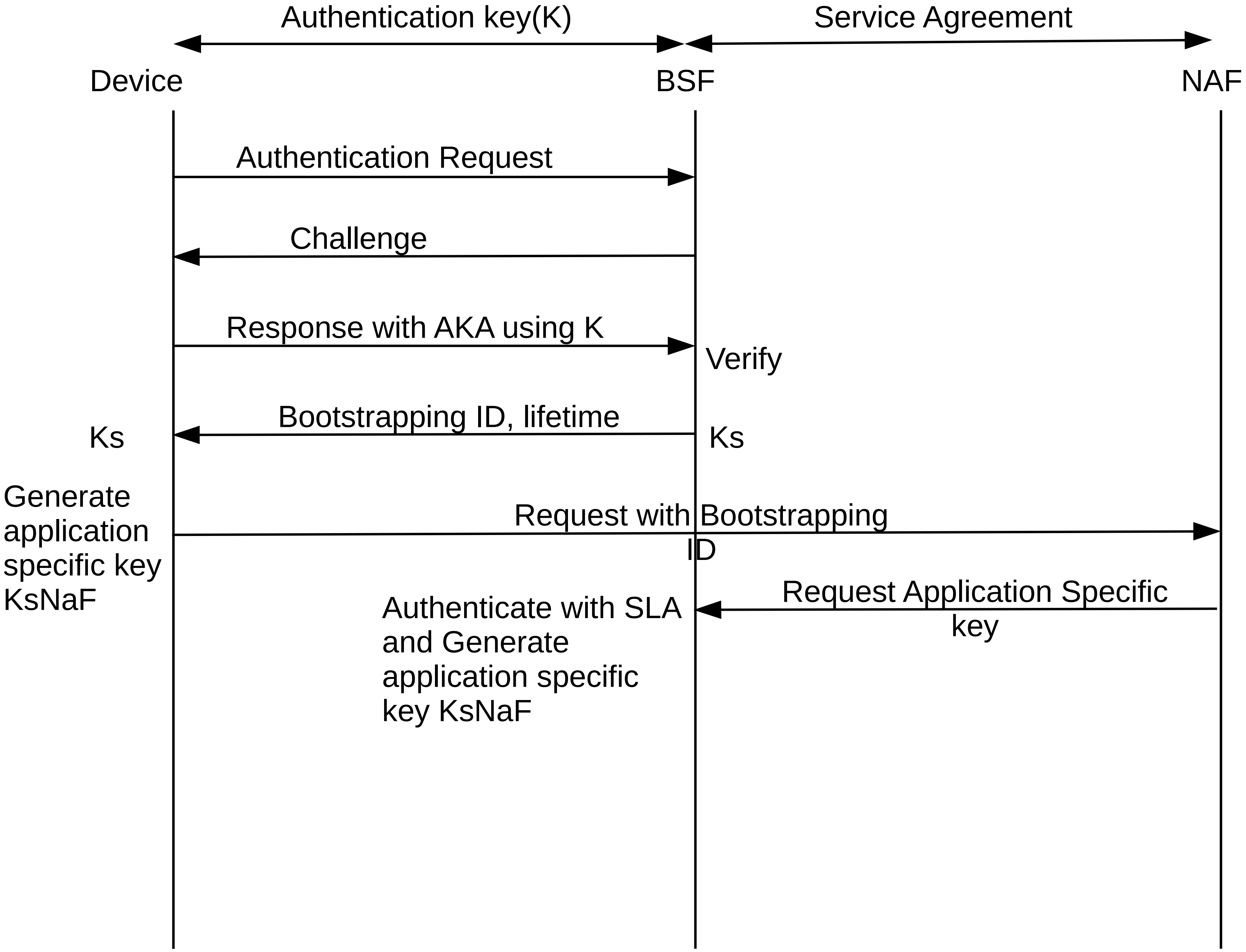

Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify

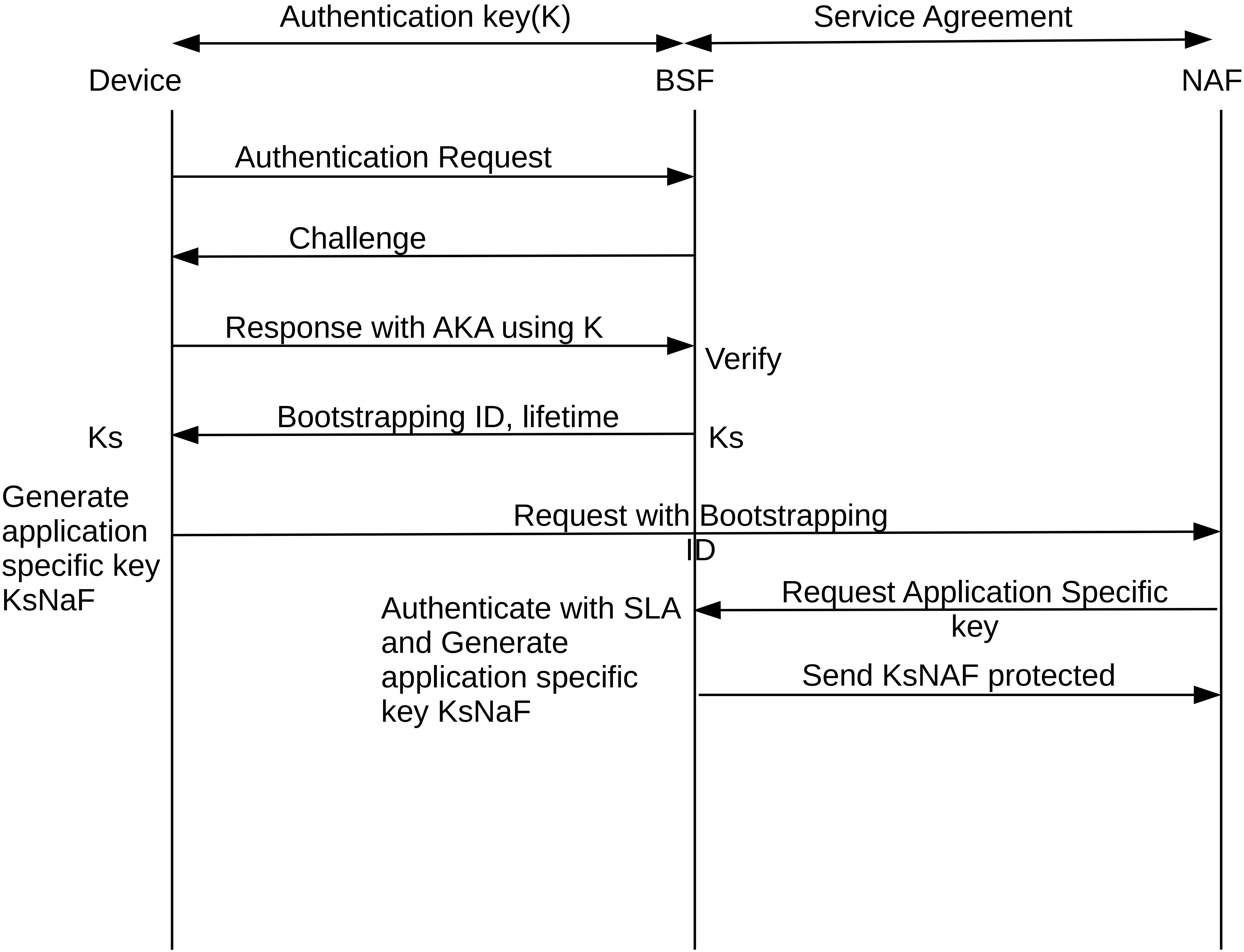

Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify
Bootstrapping ID, lifetime
Ks
Ks

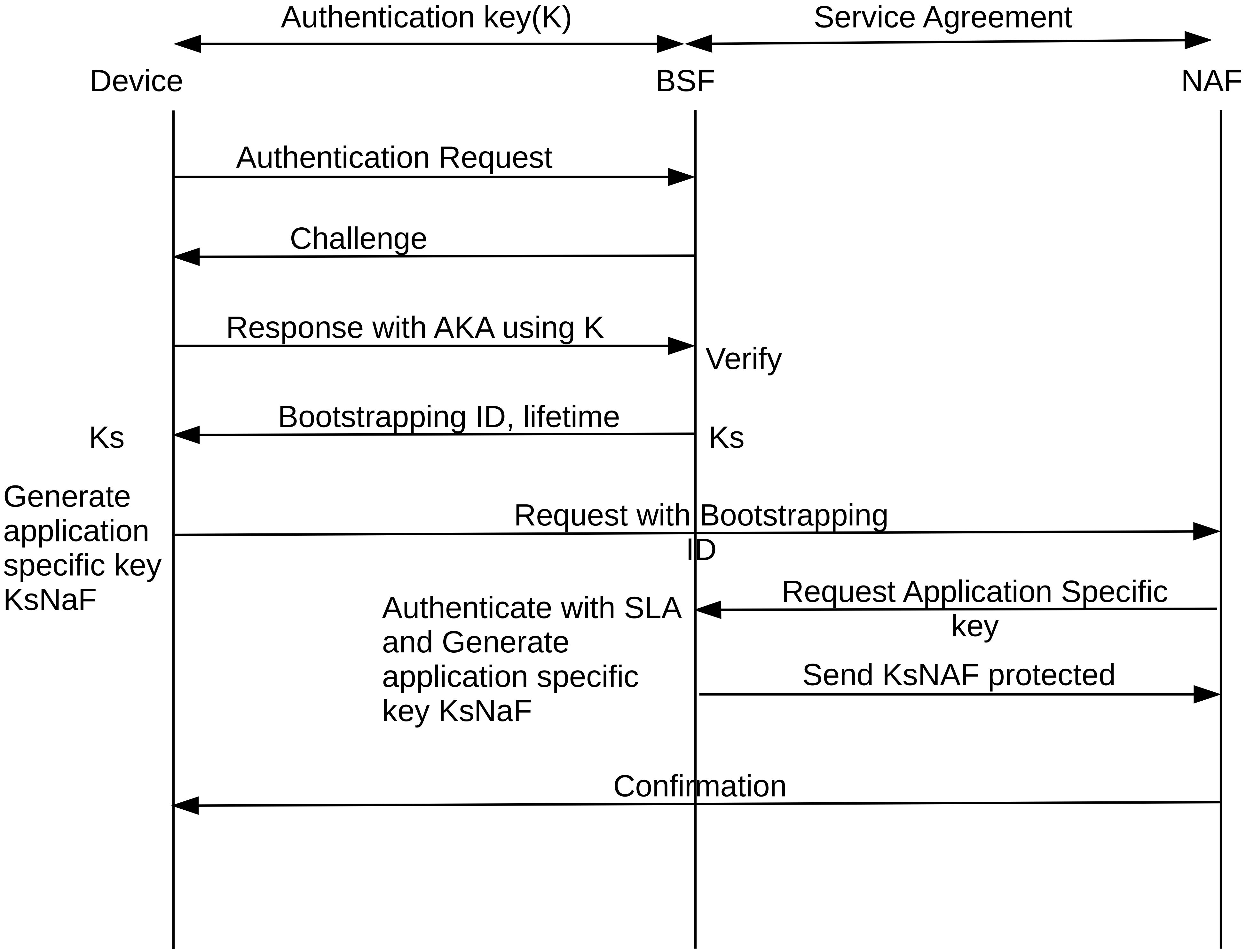

Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify
Bootstrapping ID, lifetime
Ks
Ks
Generate application specific key KsNaF

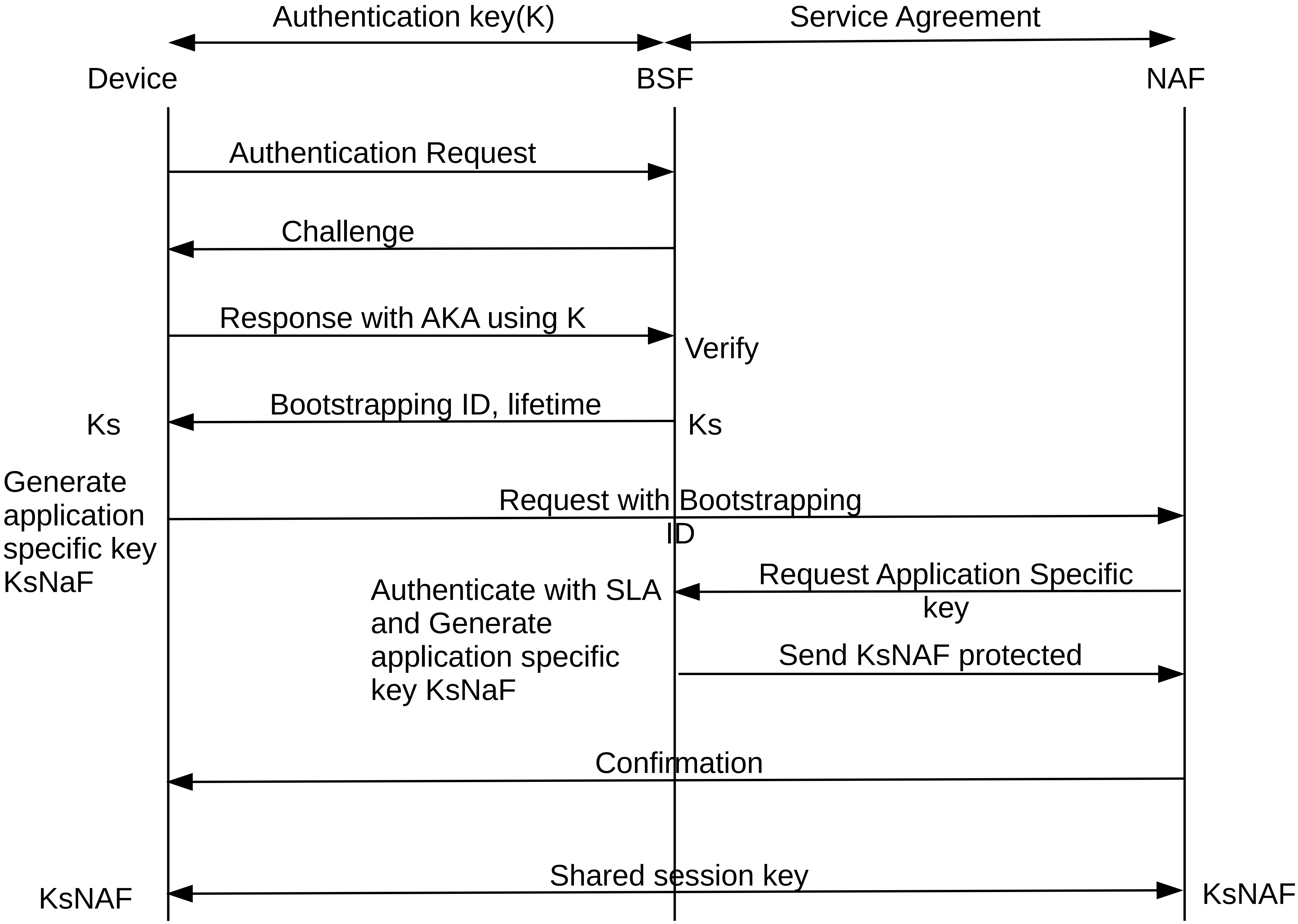

Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify
Bootstrapping ID, lifetime
Ks
Ks
Generate application specific key KsNaF
Request with Bootstrapping ID


Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify
Bootstrapping ID, lifetime
Ks
Ks
Generate application specific key KsNaF
Request with Bootstrapping ID
Request Application Specific key


Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify
Bootstrapping ID, lifetime
Ks
Ks
Generate application specific key KsNaF
Request with Bootstrapping ID
Request Application Specific key
Authenticate with SLA and Generate application specific key KsNaF


Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify
Bootstrapping ID, lifetime
Ks
Ks
Generate application specific key KsNaF
Request with Bootstrapping ID
Request Application Specific key
Authenticate with SLA and Generate application specific key KsNaF
Send KsNAF protected


Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify
Bootstrapping ID, lifetime
Ks
Ks
Generate application specific key KsNaF
Request with Bootstrapping ID
Request Application Specific key
Authenticate with SLA and Generate application specific key KsNaF
Send KsNAF protected
Confirmation


Authentication key(K)
Service Agreement
Device
BSF
NAF
Authentication Request
Challenge
Response with AKA using K
Verify
Bootstrapping ID, lifetime
Ks
Ks
Generate application specific key KsNaF
Request with Bootstrapping ID
Authenticate with SLA and Generate application specific key KsNaF
Request Application Specific key
Send KsNAF protected
Confirmation
Shared session key
KsNAF
KsNAF

Implementation Experiences

- Full HTTP stack is not needed
  - number of HTTP messages required for a GBA-run is small
  - templates

```
char httpFirstRequestFormat[82] = "GET /naf/resource HTTP/1.1\r\n"
                                   "Host: p123.example.net:8080\r\n"
                                   "Connection: Keep-Alive\r\n"
                                   "\r\n";
```

Implementation Experiences

- Resource-Constrained AES implementations are widely available
  - Gladman byte oriented AES
  - Hardware AES

Implementation Experiences

- Purging unnecessary functionality from memory after bootstrapping
  - only the session key (KsNAF) and B-TID need to be retained in the memory
  - Optionally, master key (Ks), can be retained in if connecting to multiple NAFs

Implementation Experiences

- Complete State Machine or Complex Error Handling Are Not Needed
  - Hard fail-over in most cases
  - Limit number of re-tries and increase interval between them

Implementation Details

RAM consumption	<5kB
ROM consumption	44 kB
Time for 1 GBA run	1.5 s
Energy (W =U * I * t)	150mJ
HTTP messages sent/received	8