GSS-API authentication support in SIP (work in progress)

Lars Strand
Josef Noll
Wolfgang Leister

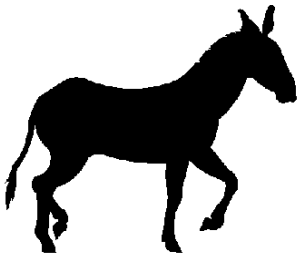
Stockholm, 3. Dec 2010

Norsk Regnesentral
NORWEGIAN COMPUTING CENTER


WANTED
Strong and flexible authentication method in SIP!

SIP authentication

1) **Digest Access Authentication (DAA) (RFC3261)**
   - mandatory
   - weak
   - widespread adoption
   - used to authenticate locally within a domain/realm



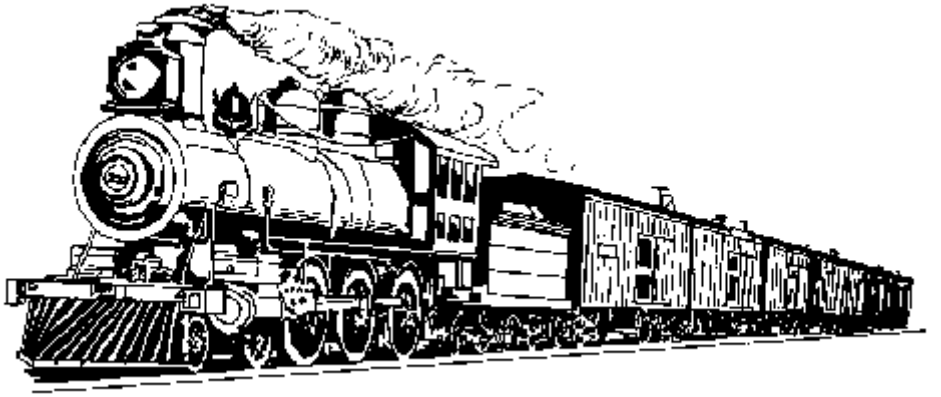
2) **S/MIME (RFC3261)**
   - Uses certificates, needs PKI = “complex and expensive”

3) **Other authentication methods**
   - P-Asserted identity (RFC3325) – in a trusted environment
   - Strong Identity (RFC4474) – using authentication service
   - Other academic approaches.

Problem

- SIP is flexible
- Problem: Different usage scenarios have different security requirements
  - Handheld devices vs. high-end SIP servers
- Goal: Modification to the SIP standard should be minimum
- Goal 2: A strong and flexible authentication methods wanted
- Solution: Add support for GSS-API

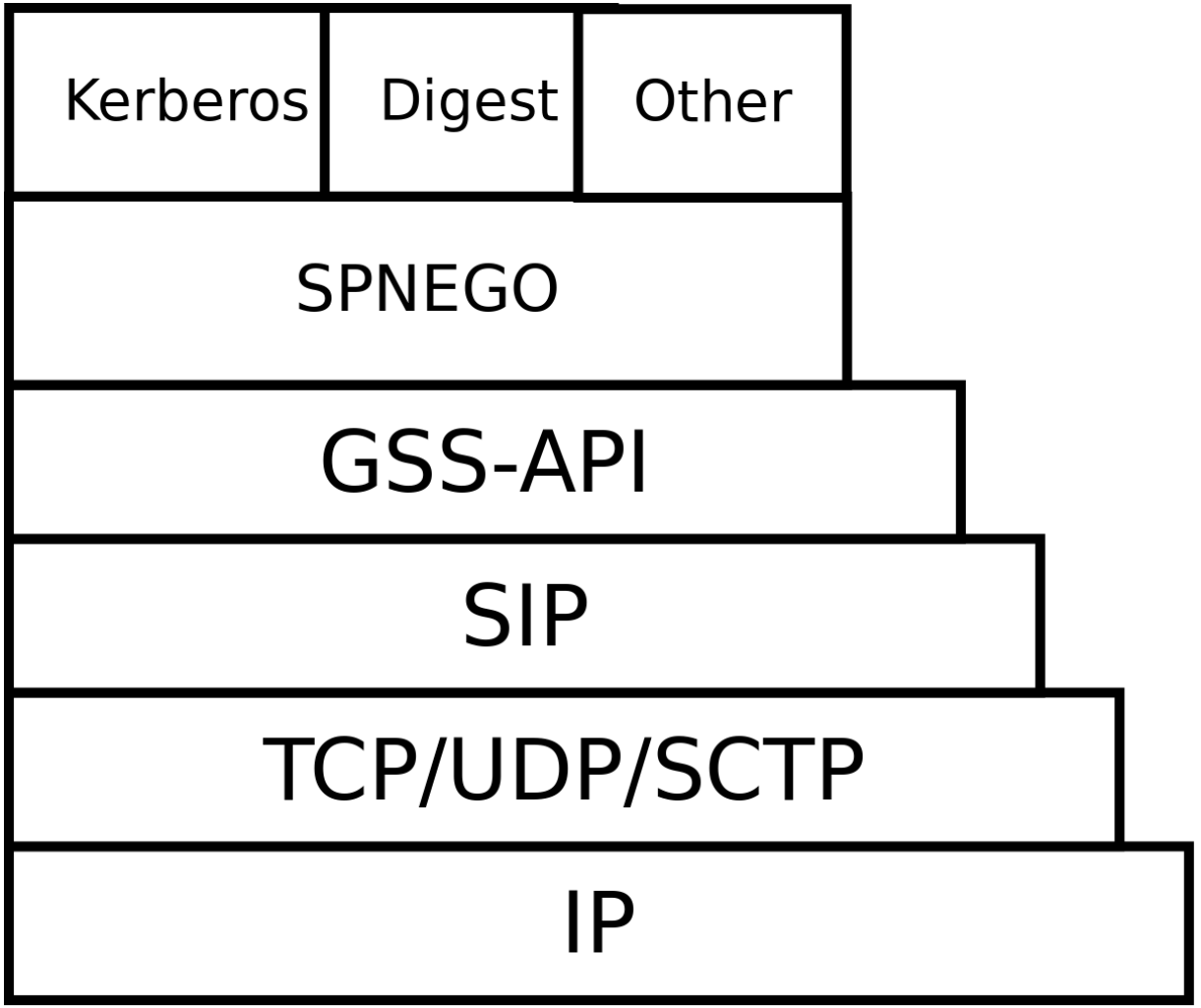
GSS-API

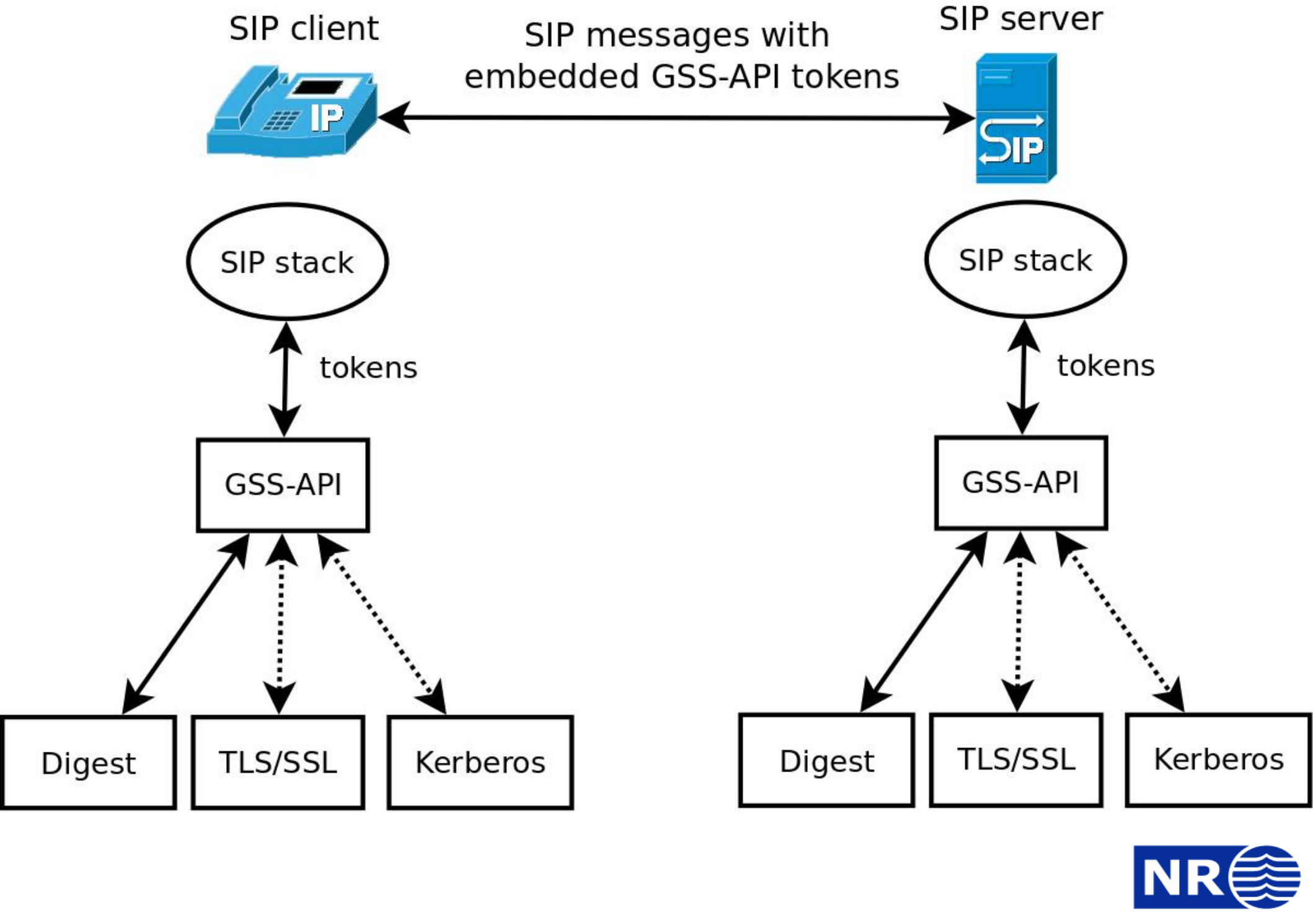


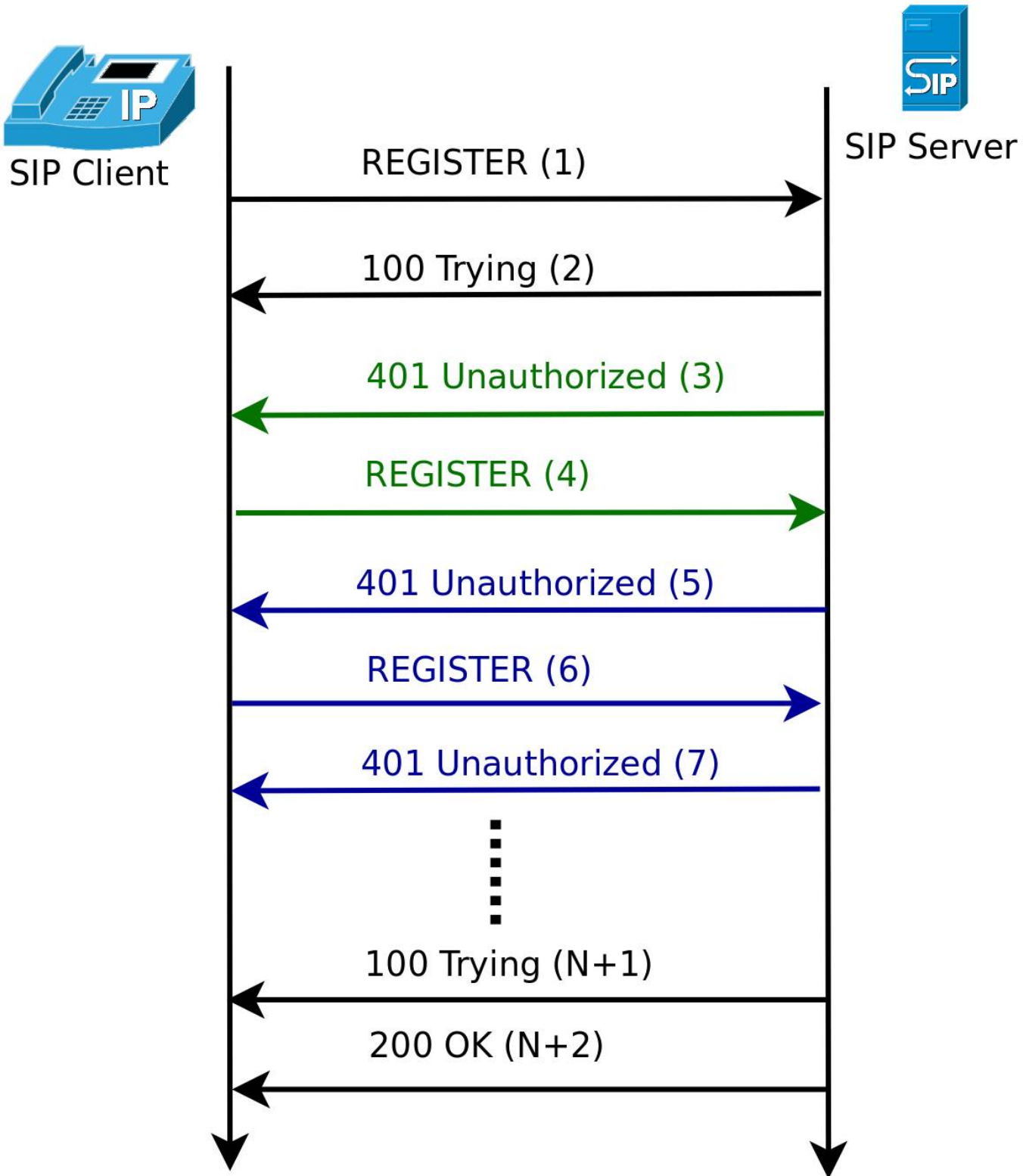
- Generic Security Services Application

  Program Interface = Interface for an application to access security services
- Mature and well-proven standard (RFC2743)
- NOT a communication protocol
  - Relies on the application (SIP) to pass data *tokens* between client and server
- Does NOT provide any security in itself
  - Relies on underlying security mechanisms
- GSS-API implementations (may) support different authentication methods
  - Digest
  - Kerberos
  - TLS
  - …
- All methods are *transparent* to the application

GSS-API stack




SIP client
SIP messages with
embedded GSS-API tokens
SIP server
IP
SIP
SIP stack
tokens
GSS-API
Digest
TLS/SSL
Kerberos
SIP stack
tokens
GSS-API
Digest
TLS/SSL
Kerberos


SIP Client
SIP Server
REGISTER (1)
100 Trying (2)
401 Unauthorized (3)
REGISTER (4)
401 Unauthorized (5)
REGISTER (6)
401 Unauthorized (7)
100 Trying (N+1)
200 OK (N+2)

SIP REGISTER

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP
    192.168.1.102;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@192.168.1.102:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: Digest
    username="alice",realm="asterisk",nonce="3b7a1395",response="ccbde1
    c3c129b3dcaa14a4d5e35519d7",uri="sip:CompanyA",algorithm=MD5
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

DAA

GSS-API token

```
1.  REGISTER sip:CompanyA SIP/2.0
2.  Via: SIP/2.0/UDP
    192.168.1.102;branch=z9hG4bK32F3EC44EB23347BFB0D488459C69E4E
3.  From: Alice <sip:alice@CompanyA>;tag=1234648905
4.  To: Alice <sip:alice@CompanyA>
5.  Contact: "Alice" <sip:alice@192.168.1.102:5060>
6.  Call-ID: 2B6449C74C10D4F95006A6C034E79E8E@CompanyA
7.  CSeq: 19481 REGISTER
8.  User-Agent: PolycomSoundPointIP-SPIP_550-UA/3.1.2.0392
9.  Authorization: GSSAPI
    token="0401000B06092A864886F712010202DACD139402AAF44350CDE32"
10. Max-Forwards: 70
11. Expires: 3600
12. Content-Length: 0
```

Conclusion

- Minimal changes to the SIP standard
- Support a range of different authentication methods
- Flexible – different implementations can support different authentication methods
- New authentication methods can be added later WITHOUT change to SIP
- Future work
  - Add an SIP extension to maintain backwards compatibility?
  - Require some authentication methods to be supported as standard? Which?
  - Vulnerable for a REGISTRATION attack?
  - Look into different GSS-API auth methods
  - Compare with SASL and SAML