SIP Authentication using EC-SRP5 Protocol

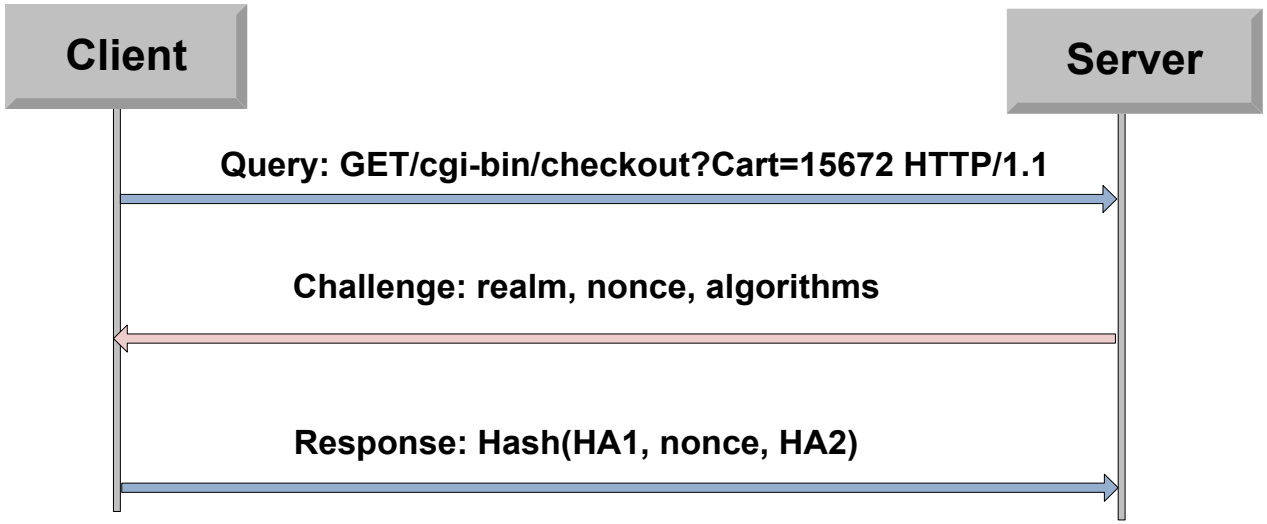
draft-liu-sipcore-ecc-srp5-00.txt

Authors: **Fuwen Liu**, Minpeng Qi and Min Zuo

SIP Authentication

-
  -
  -
-

HTTP Digest Authentication



Where: HA1=Hash (Username, realm, Passward)
HA2=Hash(Algorithms, DigistURI)

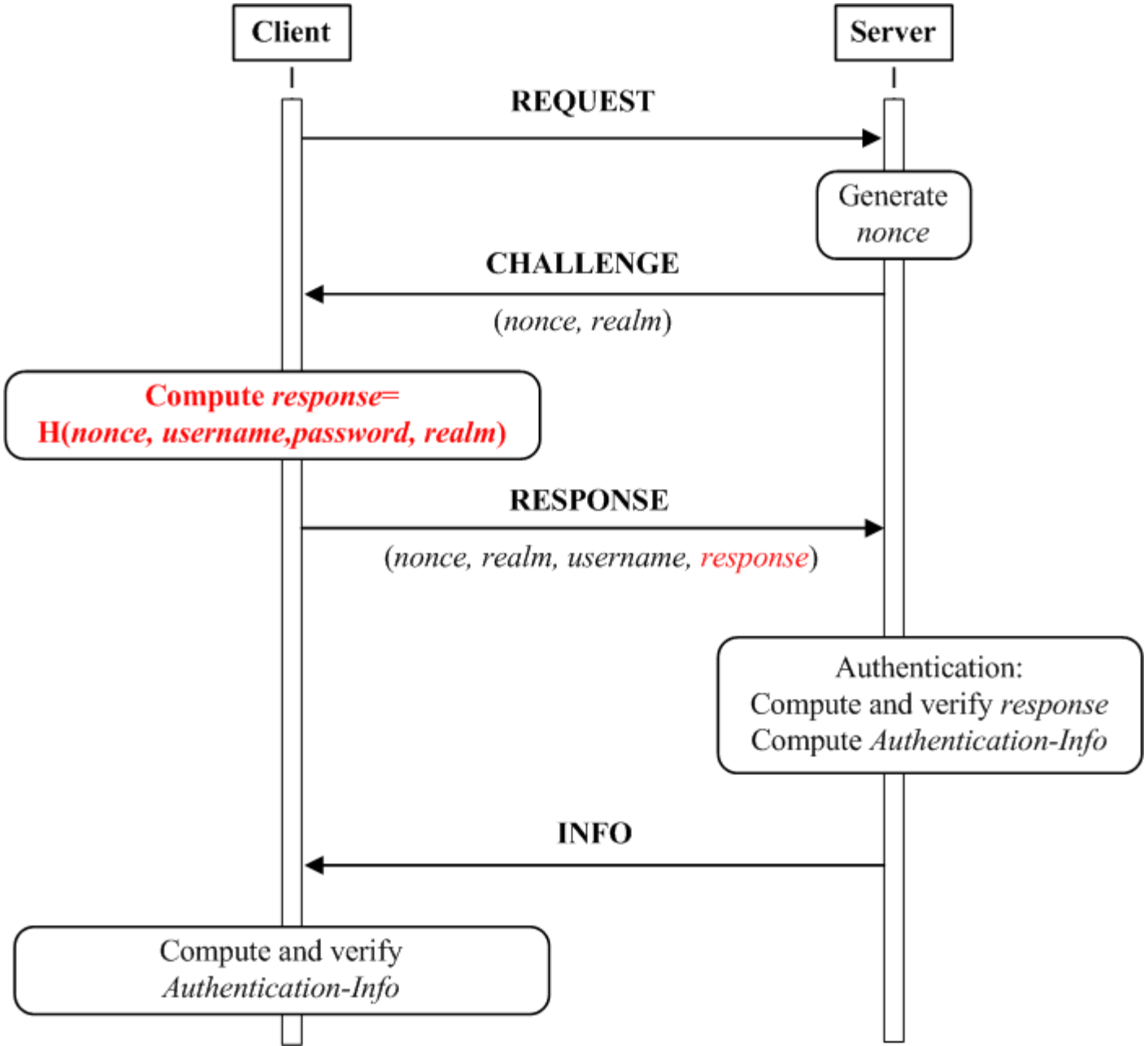
- **Breaking the scheme by computing**

Response$\stackrel{?}{=}$hash(hash(Username,realm, ***guessed Password***), nonce, HA2)

Features of Password

- Password usually short, less than 8 characters.
  - characters.

SIP Authentication based on HTTP digest



Weaknesses of SIP Authentication

- Off-line dictionary attacks are possible
  - Select a password *pw`* from password dictionary and compare

$$H(nonce, username, pw`, realm) \stackrel{?}{=} response$$

Strong Password Authentication

- **In 2009, IEEE released the standard IEEE P1363.2 regarding the password authenticated key agreement protocols**
  - **Balanced password-authenticated agreement protocols (BPKAS)**
    - Two entities know the same password and establish a shared session key
    - well suited for P2P communications
    - Three protocols are recommended: PAK, PPK, SPEKE.
    - PAK is documented in RFC 5683 as standard
  - **Augmented password-authenticated agreement protocols (APKAS)**
    - Client knows the password, while the server knows only the image of the password
    - Well suited for client/server communications
    - Seven protocols are standardized, SRP is one of representatives
    - SRP is specified in RFC 2945 by IETF

EC-SRP5 Protocol

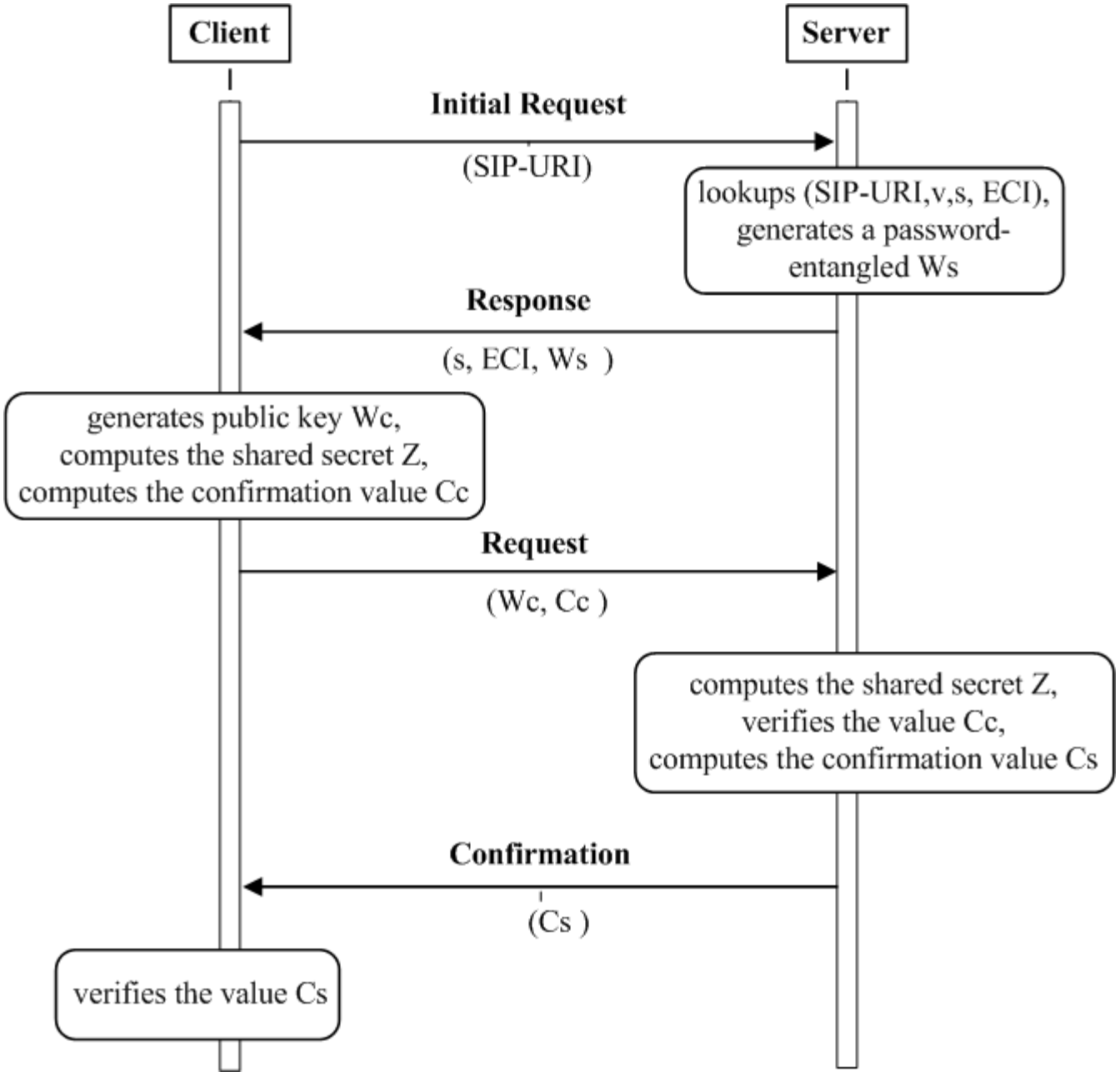
entangled into the temporary EC public key

- *To access the password, attackers have to address the ECDLP problem*

Password verifier

-
-
  -
  -
  -
  -

SIP Authentication using EC-SRP5



Security Considerations

-
  -
    -
  -
  -
-
  -
    - *when the times of authentication failure reach the default value set in advance.*

Security Considerations

- Verifying the confirmation value Cs and Cc in the client's side

- Replay attack resistance
  - Each authentication session has its unique shared secret Z
  - The client can detect the replay attack by comparing Cs with the expected confirmation value Cs‘
  - The server can detect the replay attack by comparing Cc with the expected confirmation value Cc‘

Elliptic Curve Index

SEC	ECI
secp224k1	1.3.132.0.32
secp224r1	1.3.132.0.33
secp256k1	1.3.132.0.10
secp256r1	1.2.840.10045.3.1.7
secp384r1	1.3.132.0.34
secp521k1	1.3.132.0.35

Thank you!

Discussions

- Security of the EC-SRP5 protocol
  - Although it has been documented in IEEE, its security has not been thoroughly analyzed.
- Performance of the EC-SRP5 protocol
  - Which kind of curves is efficient
- Security of ECC curves
  - Curves suggested by SEC (also recommended by NIST) are still secure ?

Appendix A: Algorithm ECPEPKGP-SRP5-SERVER

(1) Compute octet string o1=GE2OSP-X(v)
(2) Compute group element e1=ECREDP(o1)
(3) Compute group element Ws=Ts*G+e1
(4) Output Ws as the password-entangled public key

Where GE2OSP-X is used to convert group elements into octet strings. ECREDP is Elliptic Curve Random Element Derivation Primitive

Appendix B: Algorithm ECSVDP-SRP5-CLIENT

- The following steps are needed to compute the shared secret value Z in client:

(1) Compute octet string o1=GE2OSP-X(Wc)
(2) Compute octet string o2=GE2OSP-X(Ws)
(3) Compute octet string o3=SHA-256(o1|o2)
(4) compute an integer i2=OS2IP(o3)
(5) Compute octet string o4=GE2OSP-X(v)
(6) Compute group element e1=ECREDP(o4)
(7) Compute group element e2=Ws-e1
(8) Compute i3=OS2IP(SHA-256(s|SHA-256(SIP-URI|":"|Pw|ECI)))
(9) Compute group element zg= (Tc+(i2.i3))*e2
(10) Compute field element z= GE2SVFEP (zg)
(11) Compute shared secret value Z=FE2OSP (z)
(12) Output Z

Where GE2SVFEP is the primitive for group element to secret value field element conversion, FE2OSP is field element to octet string conversion primitive.

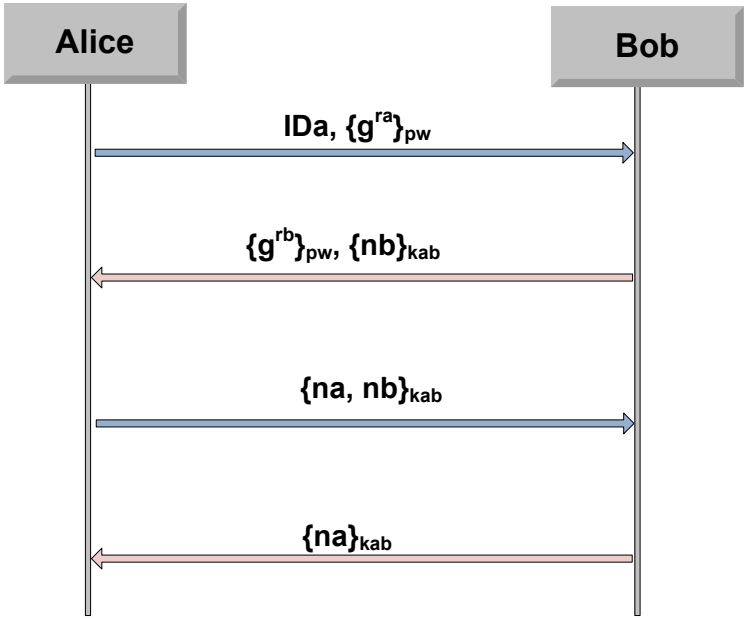
Appendix C: Algorithm ECSDVP-SRP5-SERVER

- The following steps are needed to compute the shared secret value Z in server:

(1) Compute octet string o1=GE2OSP-X(Wc)
(2) Compute octet string o2=GE2OSP-X(Ws)
(3) Compute octet string o3=SHA-256(o1|o2)
(4) compute an integer i2=OS2IP(o3)
(5) Compute group element zg= Ts*(Wc+i2*v))
(6) Compute field element z= GE2SVFEP (zg)
(7) Compute shared secret value Z=FE2OSP (z)
(8) Output Z

Encrypted key exchange-DH(EKE-DH)

- **1992, Bellovin invented EKE-DH to address this problem first. Its procedure is:**
  - Alice sends its identity *IDa* and DH-public key g^{ra} encrypted with password *Pw* to Bob
  - Bob encrypts its DH-public key g^{rb} with password *Pw*, and generates a shared $Kab=g^{ra\,rb}$. The nonce *nb* is protected by *Kab*.
  - Alice generates the shared *Kab*, and decrypts $\{nb\}_{Kab}$, and encrypts its nonce *nb* as well as *nb* with *Kab.*
  - Bob decrypts $\{na,nb\}_{Kab.}$ If the decrypted *nb* is identical to the *nb* it sended, the Alice is authenticated.
  - Alice decrypts $\{na\}_{Kab.}$ If the decrypted *na* is identical to the *na* it sended, the Bob is authenticated.



Variants of EKE-DH

- **The key point of EKE-DH is that ephemeral public DH keys are encrypted with the password.**
  - Unable to mount off-line dictionary attacks
    - Public DH keys are random strings
  - Unable to discover the session key
    - Private DH keys are unknown to attacks

- **The basic idea to combine asymmetric algorithms with symmetric algorithms to foil the off-line dictionary attacks has been extended. This can be abstracted as public DH keys are entangled by using the password. This leads to**
  - PAK (Password Authenticated key exchange) and PPK (Password Protected Key exchange)
    - Password-entangled DH public key is: $f(Pw).g^x \bmod p$
  - SPEKE (Secure Password Exponential key Exchange)
    - Password-entangled DH public key is: $f(Pw)^x \bmod p$

Standards and Patents

Protocols	Security analysis	IEEE P1363.2	RFC	Patents
EKE-DH	Several papers	--	--	US and EU patents
PAK	Provely secure	Yes	Yes	Patent held by Lucent
PPK	Provely secure	Yes	No	Patent held by Lucent
SPEKE	Provely secure	Yes	No	Phoenix held the patent
SRP	Provely secure	Yes	Yes	Standford Uni held the patent, license free
EC-SRP5	--	Yes	No	No

- **IEEE takes no position with respect to the existence of validity of any patent rights.**
- **In RFC, usually a patent-free scheme is easy to become a standard, but a patented scheme may be standardized if no patent-free scheme can replace it.**