Kerberos and weak passwords

Jacques A. Vidrine
<n@nectar.cc>
<jvidrine@verio.net>
<nectar@FreeBSD.org>

March 19, 2002

Kerberos and weak passwords

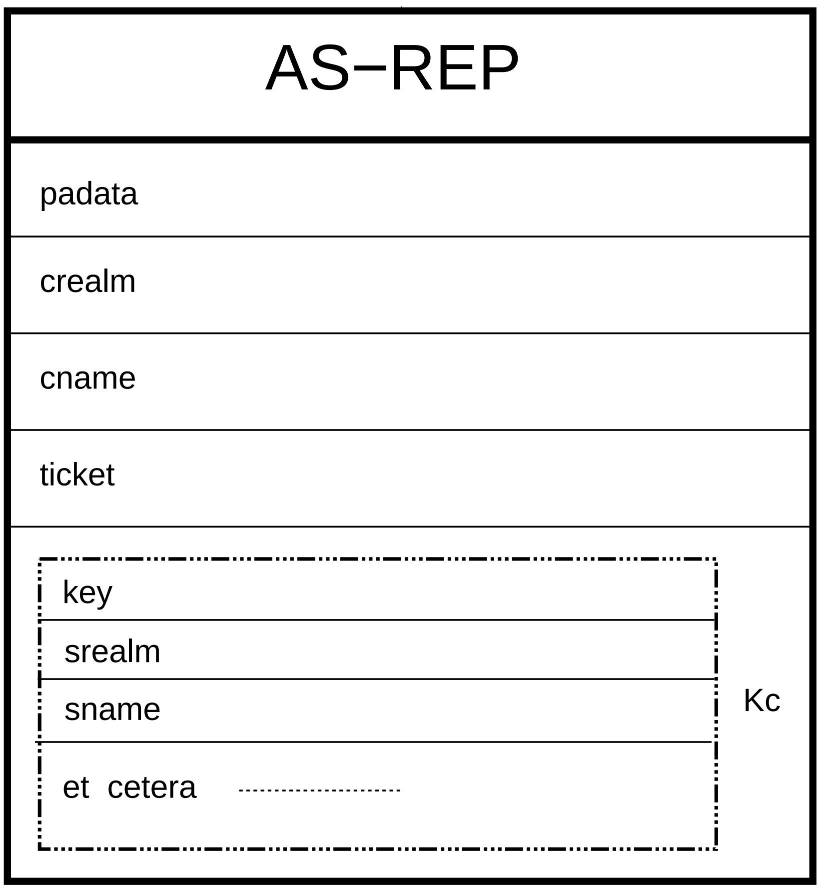
- Ignoring on-line dictionary attacks
- Opportunities for capturing ciphertext
- Usefulness of that ciphertext
- Operations for password guessing
- Rough performance numbers
- Possible solutions

Opportunities for capturing ciphertext

- AS-REP (sniff it or ask for it)
- PA-ENC-TIMESTAMP (sniff it)
- TGS-REP (ask for it)

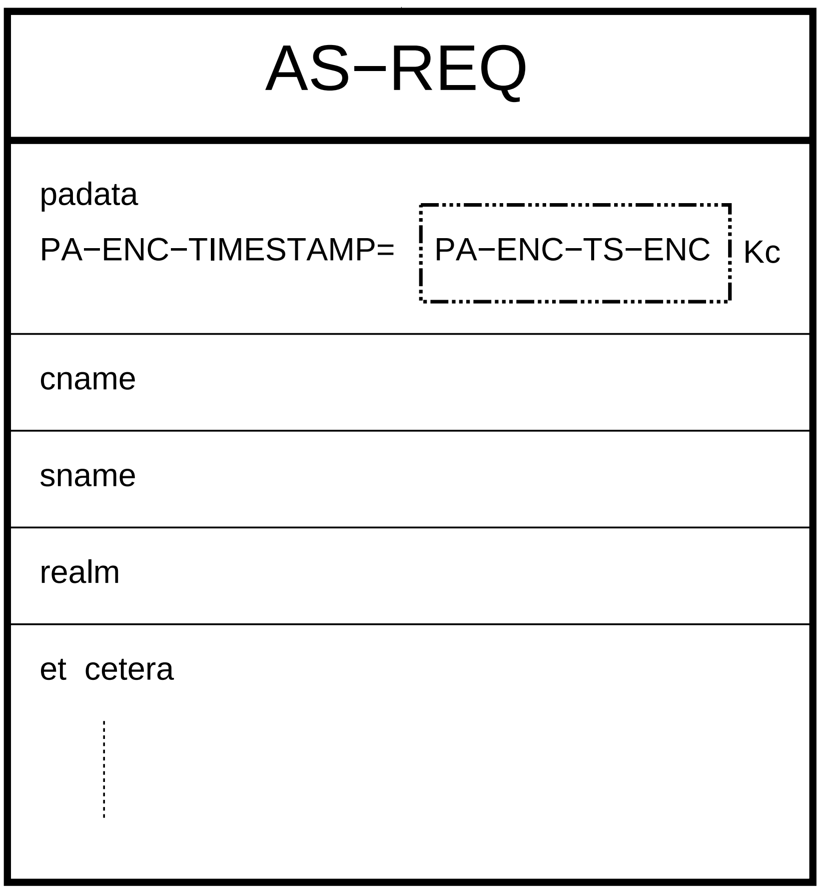
AS-REP

The AS-REP contains an EncryptionKey (and other stuff) encrypted with the client's secret.



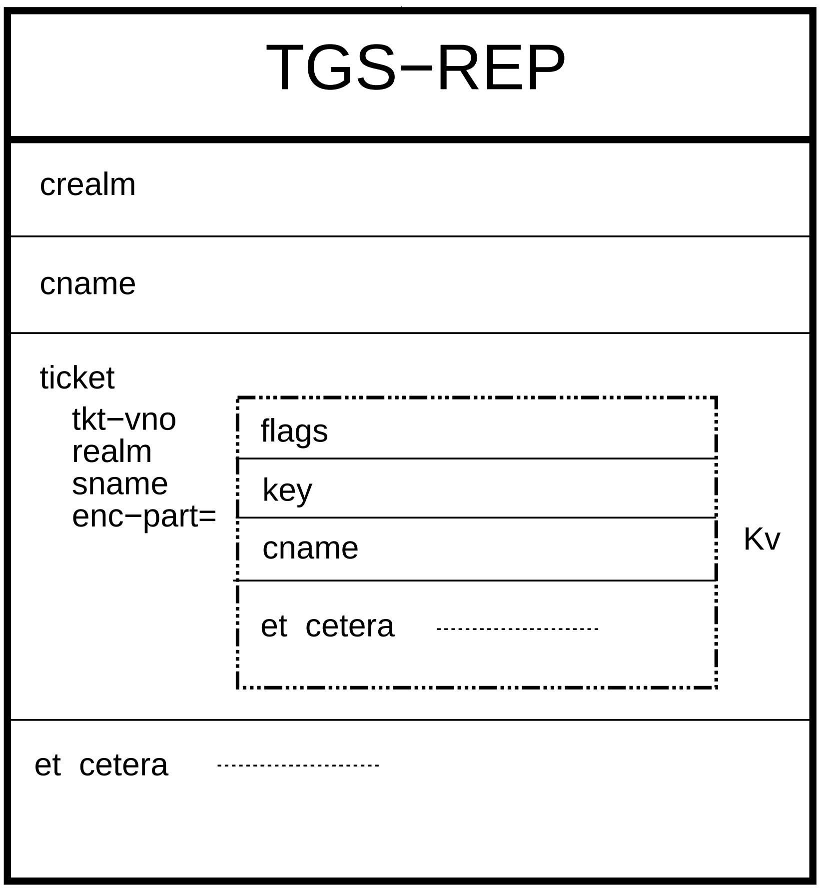
PA-ENC-TIMESTAMP

The AS-REQ may include a PA-ENC-TIMESTAMP, which is basically a KerberosTime encrypted with the client's secret.



TGS-REP

The TGS-REP contains an EncryptionKey (and other stuff) encrypted with the target's secret. This should be prevented by administratively disallowing tickets for human subjects.



Usefulness of ciphertext

ASN.1 encoding results in a regular structure to the plaintext. A simple and general approach to verifying a decryption would be to check whether the plaintext has a valid ASN.1 structure.

More specific (and probably quicker) tests can be made for each source of ciphertext, e.g. checking for the pa ttern 0xA011180F for timestamps.

Operations for password guessing

For every password / principal pair, a test requires these operations:

- String-to-key (including key derivation)
- Decryption (number of blocks according to ciphertext source and encryption type)
- Verification

Some rough numbers

Using dual-processor 1.2 GHz AMD Athlon, 1 GB RAM, decrypt PA-ENC-TS-ENC.

	des-cbc-md5	des3-cbc-sha1-kd
string-to-key	$221,587/s$	$37,299/s$
decrypt	$235,363/s$	$146,888/s$
verify	$485,858/s$	
total	$92,423/s$	$28,030/s$

Possible solutions

- DCE RFC 26.0-like
- SSL/TLS
- LEAF
- SRP / PDM

DCE RFC 26.0, SSL/TLS, and LEAF all introduce key distribution / management issues for clients that did not previously exist.

Password Derived Moduli (PDM)

Cool, and potentially enables a two message exchange. But:

- SACRED WG dropped PDM in favor of SRP.
- IP Storage WG seem to favor SRP (although presently debated down the hall).
- PDM client performance poor ‘by design’.
- At least Stanford has provided an IPR statement.

Secure Remote Password (SRP)

SRP strawman. RFC 2945. g and N are well-known.

- C->KDC: AS-REQ
- KDC->C: KRB-ERROR PREAUTH-REQUIRED, salt, $B = (v + g^b)$ mod N, R
- C->KDC: AS-REQ PA-DATA, $A = g^a$ mod N, $E_K(\text{SHA1}(MD))$
- KDC->C: AS-REP $E_K(\text{enc-part})$

Other ideas

Use (a) SACRED protocols; (b) AS exchange; or (c) new message exchange to obtain a long-term high-quality secret to then use in the 'real' AS exchange.

- Advantage: the KDC is not required to keep state as in four message SRP strawman.

PDM enables a two message protocol for the AS exchange, but (a) performance on the client is poor; and (b) at least one other working group decided that PDM was riskier than SRP from an IPR point-of-view.

Can SRP be modified such that the user's password is committed to in the first message? Need a real cryptographer.

What now?

- Internet draft for Kerberos PA-SRP or whatever
- I'm a newbie and would like assistance