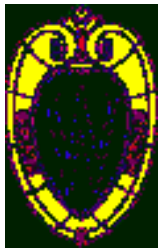


CS/ECE/Math 435
Introduction to Cryptography

Professor Chris DeMarco
Department of Electrical & Computer Engineering
University of Wisconsin-Madison

Spring Semester 2021

RSA-based password/authentication protocol.

- User requesting access: entity "A"
  Centralized entity managing access: "B"

- Consistent with RSA system, A chooses primes p, q, with $N = p \cdot q$, and $\varphi = (p-1) \cdot (q-1)$. A further chooses integers d, e such that $\text{mod}(d \cdot e, \varphi) = 1$.

- (N, e) are public; d, p, q remain internally secret to A. A's identity tied to A publishing (N, e).

  Recall, based on above, RSA defines

  Encryption function $E(x) = \text{mod}(x^e, N)$
  (uses public information)

  Decryption function: $D(y) = \text{mod}(y^d, N)$
  (uses information internally secret to A)

- User A sends access request to B;
- B generates a randomly selected message "y", sends y to A.
- A computes $x = D(y)$, sends x back to B
- B tests $y \stackrel{?}{=} E(x)$

(yes $\Rightarrow$ A confirmed associated to (N, e))

Alternatives / Supplements to password access : "keyed" client-server interactions.

In modern networks, many communication processes benefit from a secure protocol, but the volume and frequency of these communications make user intervention (e.g., distinct password entry) impractical.

A commonly used approach employs time-stamped, limited duration keys.

Kerberos Protocol

- Communication session desired between users U and V, facilitated by "trusted" third entity (e.g., network server), T
- U and V assumed to have previously established secure communication with T.

- Recognize motivation - we want to enable many automated V-V interactions, without human user intervention.

But we assume V-T, V-T interactions are longer duration. We can afford more thorough (perhaps human-involved) password authentication when V and V first join the network.

- Each U-V communication session will have a beginning timestamp, t, and a duration or lifetime, l.

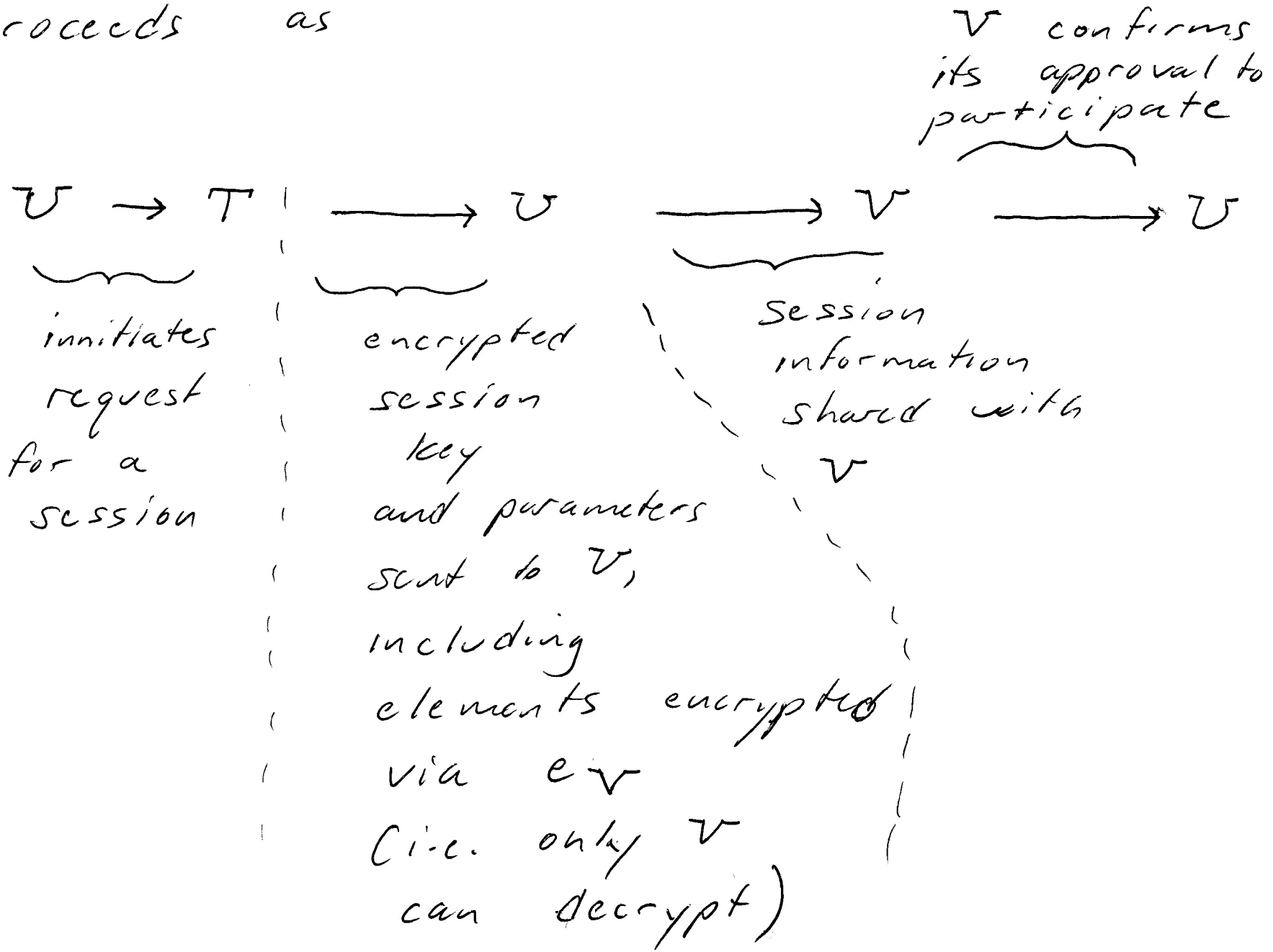
- U, V, and T all use a common symmetric, secret key encryption scheme, such as DES.

- U-T already has established a shared key, and can encrypt/decrypt. Denote these e_U, d_U.

- V-T already has shared key, similarly yielding e_V, d_V

- Kerberos protocol is structured so that only U has to interact directly with T to request session (but U must identify V in its request to T, and structure ensures V is "willing")

So communication sequence proceeds as



Bach notes provide a nice illustrative example, substituting "toy" Vigenere cipher for the DES or AES used in actual practice.

"Characters" of all messages are $\mathbb{Z}_{10}$, with elements of messages composed of:

i) User ID's of two "characters" (i.e., two integes, each $\in \{0, 1, \ldots 9\}$, so user ID's rang 00 to 99)

ii) Similarly, keys are two characters from $\mathbb{Z}_{10}$, 00 to 99;

iii) Time stamps two characters from $\mathbb{Z}_{10}$, 00 to 99;

iv) Lifetime represented by one character from $\mathbb{Z}_{10}$, 0 to 9;

Initialization:

User U has ID 05, U-T key 15

User V has ID 06, V-T key 19

1) First $U \rightarrow T$ message

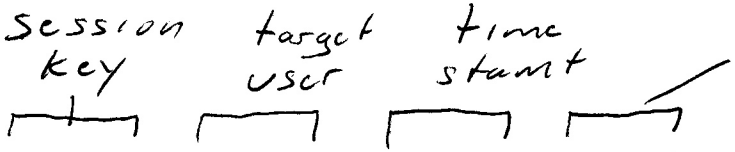
"User 05 requests communication session with user 06"

(this can be unencrypted).

2) Suppose (1) reaches T at "time" 20. T takes timestamp 20, assigns lifetime 5, assigns session key 34.

The information above should be communicated securely from T back to U, so it is encrypted.

Message order/convention used here



plaintext
7 characters

[34 06 20 5]

Periodic 2 Vigenere cv [15 15 15 1] (would continue with 5 if message had more characters)

(just repeat 15)

"character-by-character" encryption in $\mathbb{Z}_{10}$

---

m_1

ciphertext to be sent from T back to U.

[4 9 1 1 3 5 6]

But T also creates a second ciphertext from similar plaintext, using e_V:

but here, user ID 05

plaintext: [3 4 05 20 5]

periodic cipher [1 9 1 9 19 1]

---

ciphertext message: [4 3 1 4 3 9 6]

m_2, to be sent from T back to V

3) U receives ciphertexts m_1, m_2 from T.

U decrypts m_1, $d_U(m_1)$

$= [\ 49 \quad 11 \quad 35 \quad 6\ ]$

$- [\ 15 \quad 15 \quad 15 \quad 1\ ]$

$\boxed{34}\ 06\ 20\ 5$

↓

U now has the session key

U creates ciphertext message, using encryption with session key, composed of user ID and timestamp

$$m_3 = e_s([05 \quad 20])$$

$$= [05 \quad 20]$$

$$+ [34 \quad 34]$$

$$= [39 \quad 54]$$

4) V receives m_2 (i.e., V just forwards m_2 from T to V) and m_3 from U.

V decrypts m_2: $d_V(m_2)$

$$= [\ 43 \quad 14 \quad 39 \quad 6\ ]$$

$$- [\ 19 \quad 19 \quad 19 \quad 1\ ]$$

$$= [\ \boxed{34} \quad 05 \quad 20 \quad 5\ ]$$

↑

Now V has the session key

V can now decrypt m_3,

$d_S(m_3)$

$= [39 \quad 54]$

$- [34 \quad 34]$

$= [05 \quad 20]$

By convention, Kerberos requires V to increment time stamp by 1, and communicate this back to V as a last consistency check

5) V computes $21 = 20 + 1$,

encrypts this to create $m_4 = e_{S'}([21])$

$$= [21]$$
$$+ [34]$$
$$\underline{[55]}$$

this m_4 is sent from V to U

6) U decrypts m_4, checks for consistency with original timestamp. If yes, session proceeds using e_S, d_S.

Adapting RSA to provide digital Signature functionality.

General goals of a digital signature function:

Without prior exchange of a secret key between document author and document reader, we wish to provide the reader with:

1) Confirmation of author identity (preventing another entity from falsely claiming authorship).

2) Confirmation that document retains the exact content it had at signing (preventing modifications to document, post-signature)

---

3) Confirmation of time/date of signing - but this is typically encompassed by (2), provided that part of the document content is a verifiable timestamp.

RSA-based techniques focus on (1) & (2).

Assumption for treatment here:

Document / message to be signed "fits" within one block of the RSA encryption scheme. Recall that RSA block size in bits is set by selected prime number parameter "p".

Subsequent lectures will examine use of hash functions to relax the assumption above, enabling RSA methods to provide digital signature on longer documents / messages.

RSA Digital Signature Protocol

Message author/signer first selects large primes p, q, setting $N = p \cdot q$, and $\varphi = (p-1) \cdot (q-1)$. Next selects integers d, e, such that $\text{mod}(d \cdot e, \varphi) = 1$. Signer publishes (d, N); p, q, e remain internally secret to signer.

A signed message is pair (x, y), where x is original message, $y = \text{mod}(x^e, N)$.

Reader wishing to verify message tests:

$$\mathrm{mod}(y^d, N) \stackrel{?}{=} x$$

Again, test above really just confirms consistency of x and y with public key (N, d), confirming that $y = \mathrm{mod}(x^e, N)$. But reader must have trust that the identity of the signer really is linked to the public key (N, d).