Chapter 14

Entity Authentication

Entity Authentication

Entity authentication is a technique designed to let one party prove the identity of another party. An entity can be a person, a process, a client, or a server. The entity whose identity needs to be proved is called the claimant; the party that tries to verify the identity of the claimant is called the verifier.

Data-Origin Versus Entity Authentication

There are two differences between message authentication (data-origin authentication), and entity authentication.

1) ***Message authentication might not happen in real time; entity authentication does.***
2) ***Message authentication simply authenticates one message; the process needs to be repeated for each new message. Entity authentication authenticates the claimant for the entire duration of a session.***

Verification Categories

Something known

Examples: PIN or passowrd

Something possessed

Examples: Id card, smart card (contact or contactless)

Example: contactless credit card

a.k.a. payWave card or No-swipe card

Something inherent

Examples: voice, retinal pattern, handwriting

Passwords

The simplest and oldest method of entity authentication is the password-based authentication, where the password is something that the claimant knows.

Types of Passowrd

Fixed Password: ***can be use over and over again for every access.***

One-Time Password: ***is used only once. This type of password makes eavesdropping useless.***

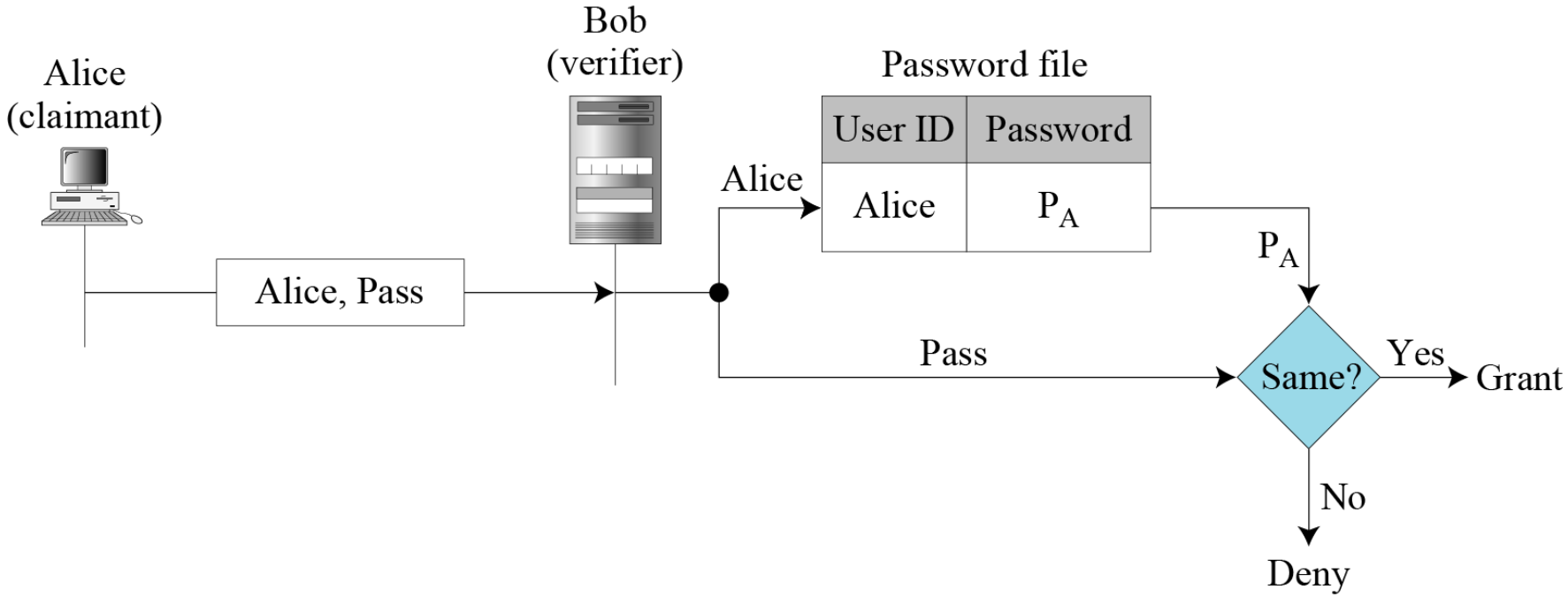
Fixed Password

First Approach

The system keeps a table (file) of two columns: User ID and password in plaintext. The users who wants to access the system sends his user ID and password in plaintext. If the password stored in the table matches the one provided by the user, access is granted.

P_A: Alice's stored password

Pass: Password sent by claimant



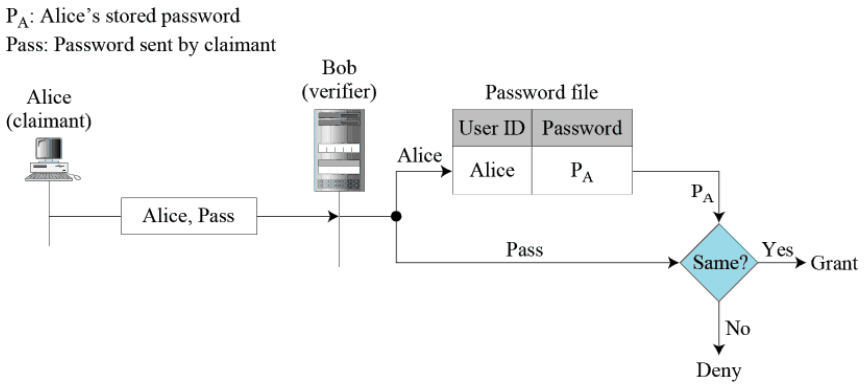
Attacks on the First Approach:

Eavesdropping: **Eve can listen to the line and intercept the message containing the unprotected password.**

Password Stealing: **the adversary can physically steal the password if it is written down on some paper or document.**

Accessing the password file: **the adversary can hack into the system and get access to the password file.**

Guessing (brute force attack): **the adversary can try to guess the password.**



Fixed Password

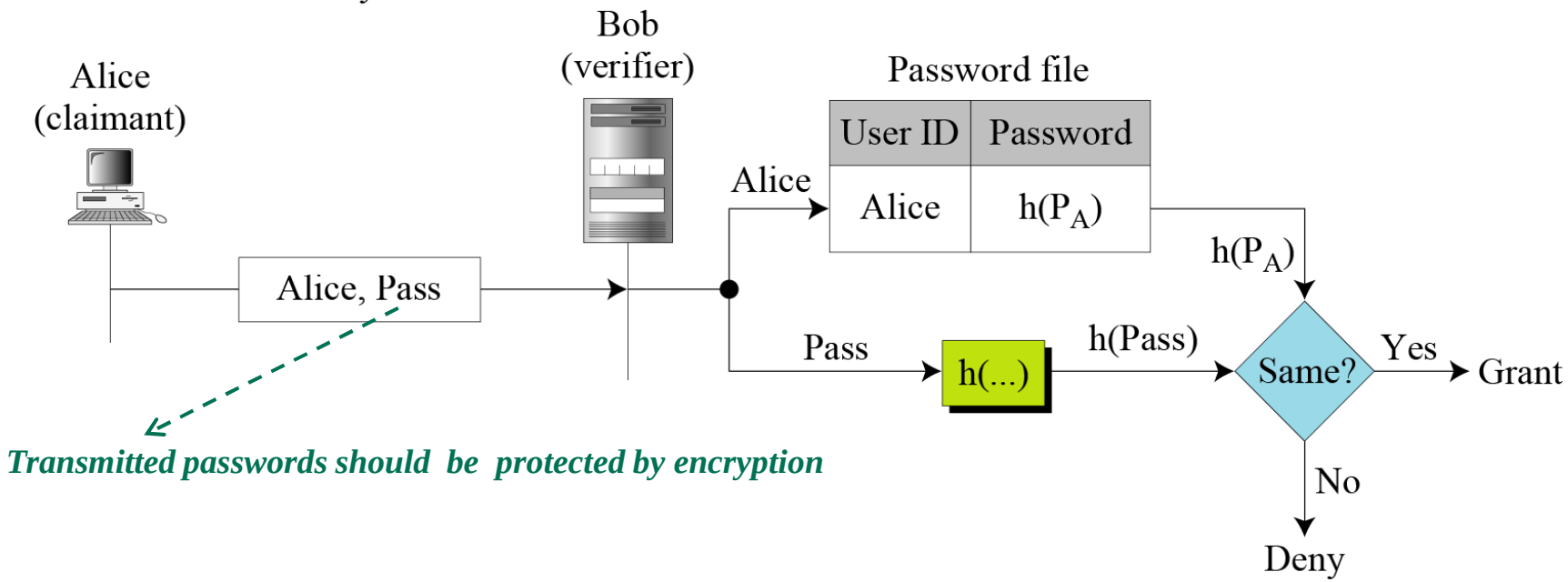
Second Approach

The password file stores the hash of the password instead of the plaintext password. Because the hash function is a one-way function, users that can read the contents of the file cannot know the value of the password.

Dictionary Attack: the adversary tries to find one password in the table regardless of the user ID. The adversary applies the hash function to randomly created passwords and searches the second column of the password file to find a match.

P_A: Alice's stored password

Pass: Password sent by claimant



Fixed Password

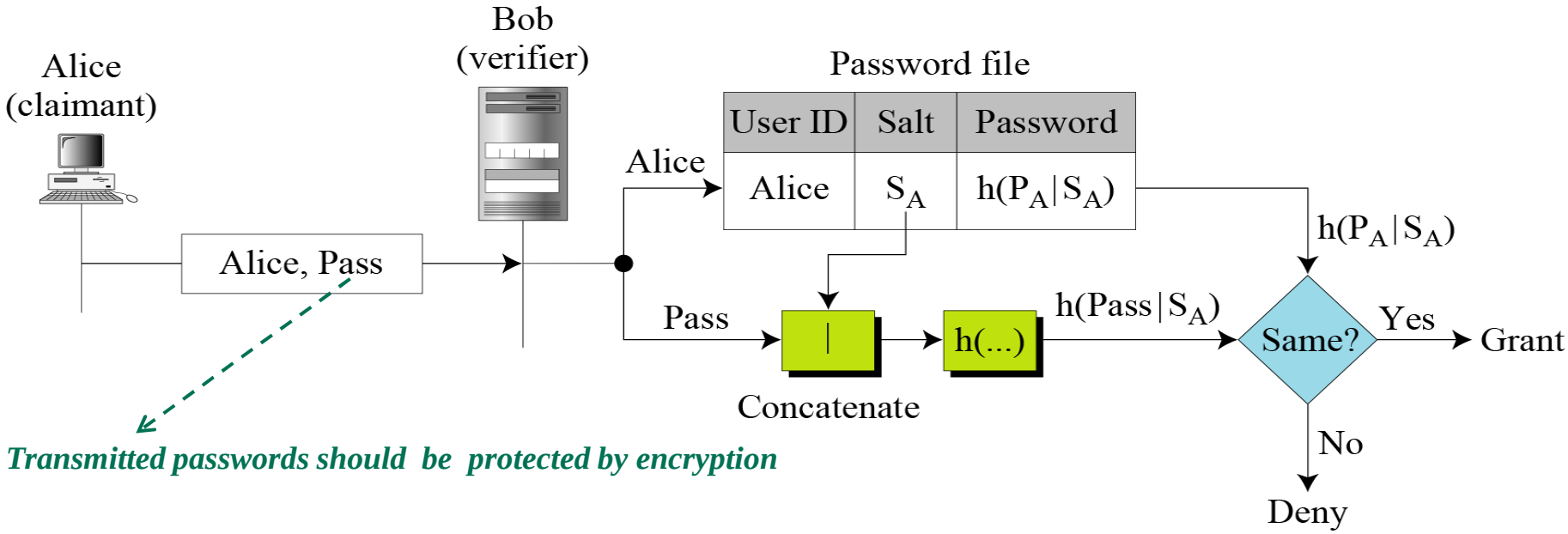
Third Approach

Salting the password: ***A long random string, called the salt, is concatenated to the password before applying the hash. The user ID, the salt and the hash are stored in the password file. If the Salt field is read-protected, dictionary attacks become more difficult. Unix OS uses a variation of salting.***

P_A: Alice's password

S_A: Alice's salt

Pass: Password sent by claimant



One-Time Password

First Approach

In the first approach, the user and the system agree upon a list of passwords; each password can be used once.

Second Approach

In the second approach, the user and the system agree to sequentially update the password.

Third Approach (Leslie Lamport)

In the third approach, the user and the system create a sequentially updated password using a hash function. The scheme is based on one-way hash chains. The notation h^n means applying the hash function n times.

$$h^n(x) = h(h^{n-1}(x)) \quad h^{n-1}(x) = h(h^{n-2}(x)) \quad \ldots \quad h^2(x) = h(h(x)) \quad h^1(x) = h(x)$$

One-way Hash Chain

- **The hash chain has a length n and can be used to generate n one-time passwords.**
- **The passwords are derived by repeatedly applying the hash function on an initial secret s, selected by the user.**
- **Due to the pre-image resistance property of the hash function (e.g., SHA-1), the hash values P_i are distinct and can be used to represent n one-time passwords.**

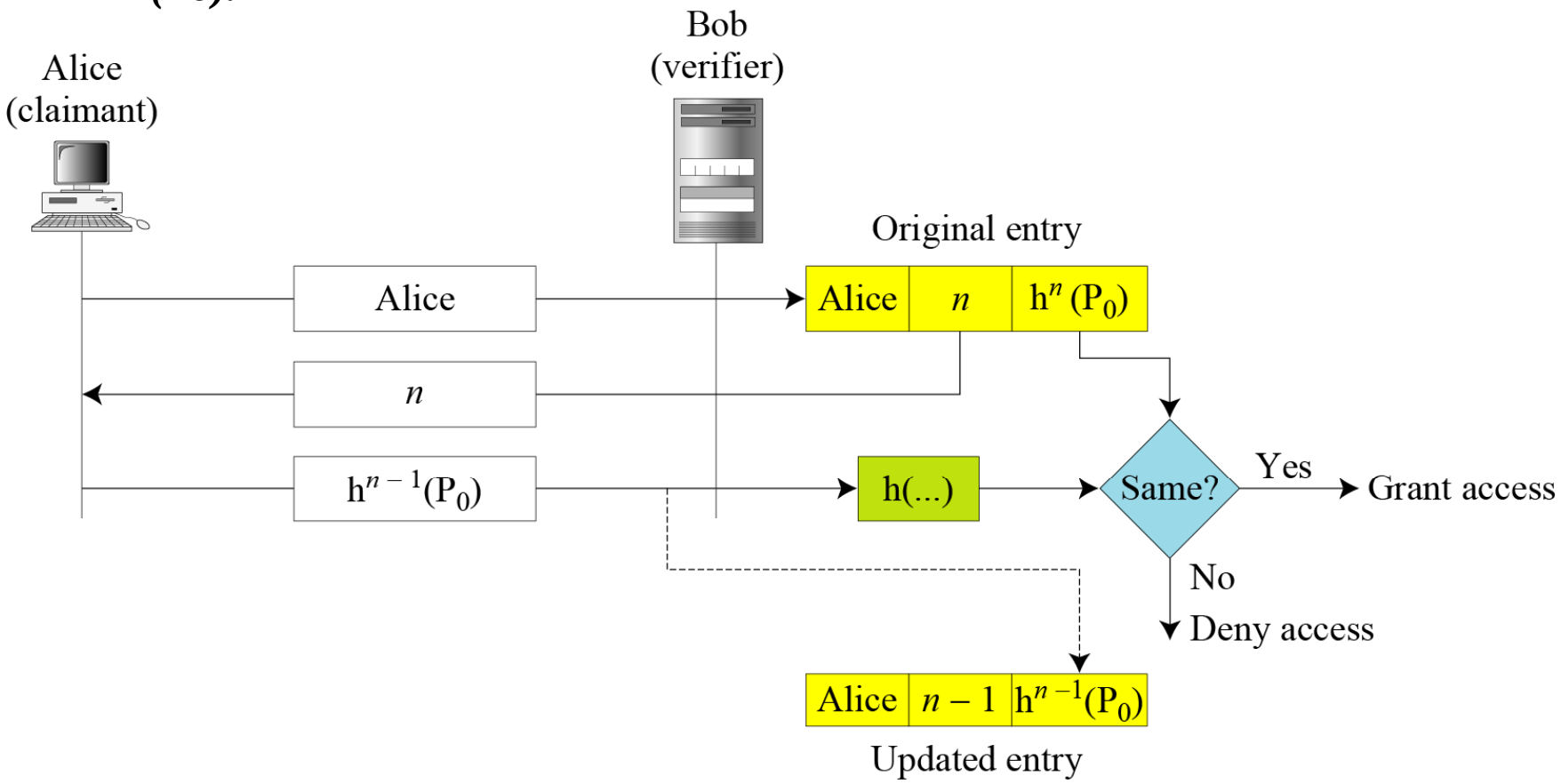
$P_0 = s$ // initial password selected by user
$P_1 = h(P_0) = h^1(s)$
$P_2 = h(P_1) = h^2(s)$

…
$P_j = h(P_{j-1}) = h^j(s)$
…
$P_{n-1} = h(P_{n-2}) = h^{n-1}(s)$
$P_n = h(P_{n-1}) = h^n(s)$ $\rightarrow P_n$ **must be the first transmitted one-time password**

- **To make the system cryptographically strong and prevent attacks, the passwords must be transmitted in *reverse order*, i.e., the first password transmitted must be $P_n = h^n(s)$, not $P_1 = h(s)$. This is because if h(s) is used as the first password, the attacker can use it to predict all future passwords by repeatedly applying the hash function on the intercepted password P_1.**

Lamport one-time password

In this scheme, the user can access the system n times before a new setup is needed. The system stores an initial entry containing the value n and the hash $h^n(Po)$, *where Po* is the initial password chosen by the user. Each time the user accesses the system successfully, the value of n in the stored entity is decremented and the stored hash $h^n(Po)$ is replaced with the new hash $h^{n-1}(Po)$.



Challenge-Response

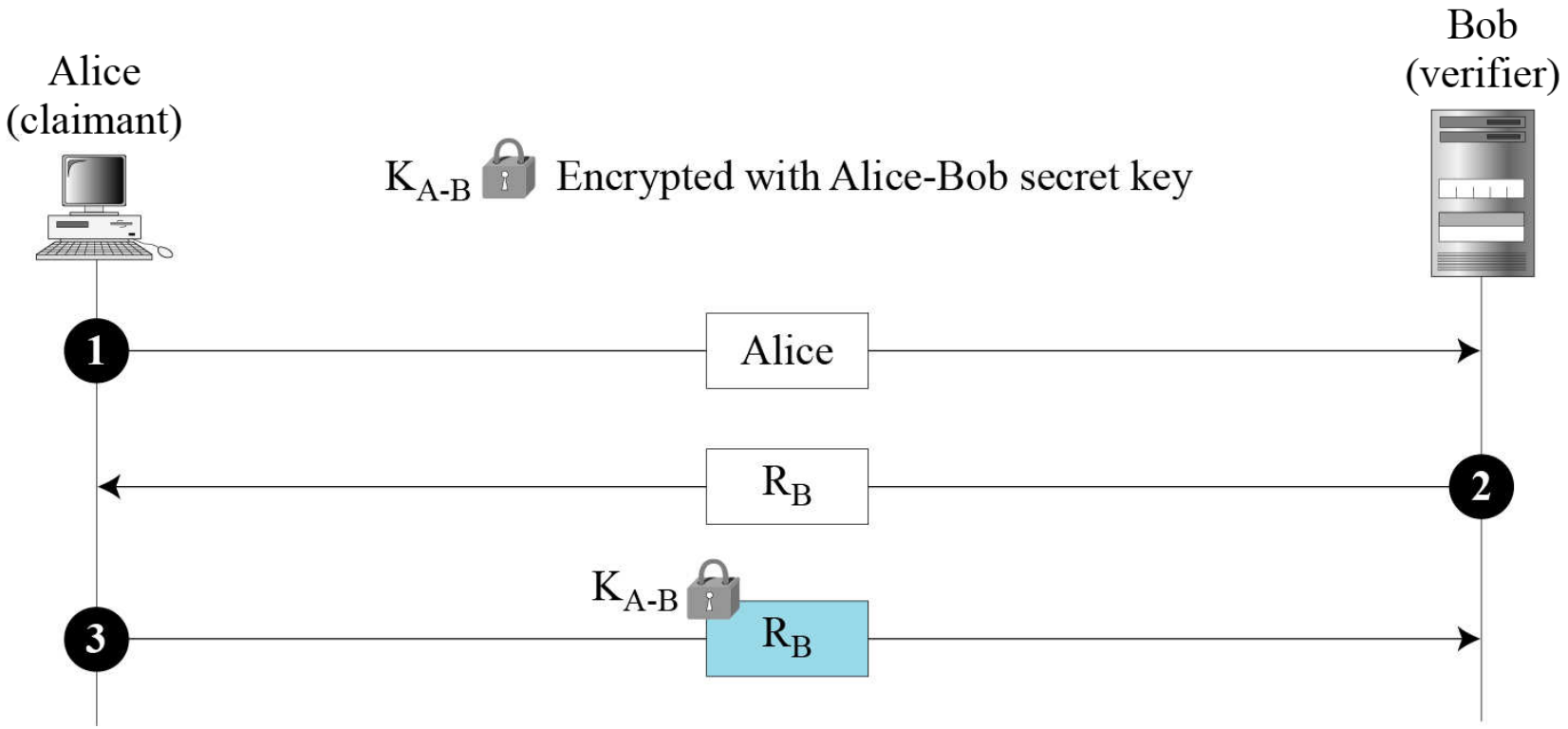
In password authentication, the claimant proves his identity by demonstrating that he knows a secret, the password. In challenge-response authentication, the claimant proves that he knows a secret without sending it.

The challenge is a time-varying value sent by the verifier; the response is the result of a function applied on the challenge.

Using a Symmetric-Key Cipher

Symmetric key with Nonce Challenge

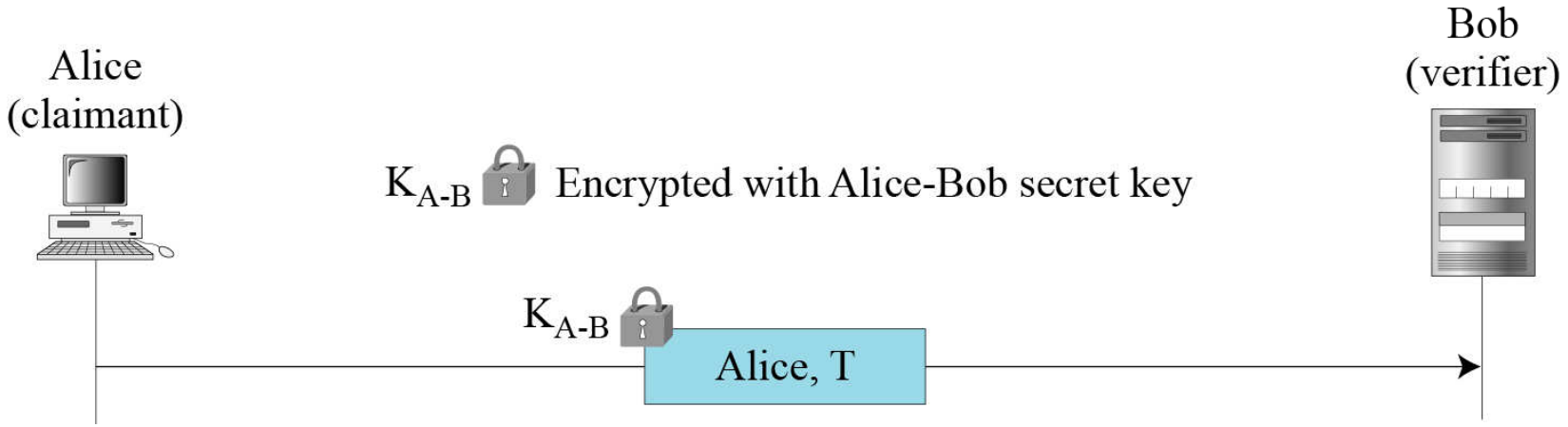
The verifier sends a nonce and the claimant responds to the challenge using the secret key shared between the claimant and the verifier. The use of nonce effectively prevents replay attacks.



Using a Symmetric-Key Cipher

Symmetric key with Timestamp

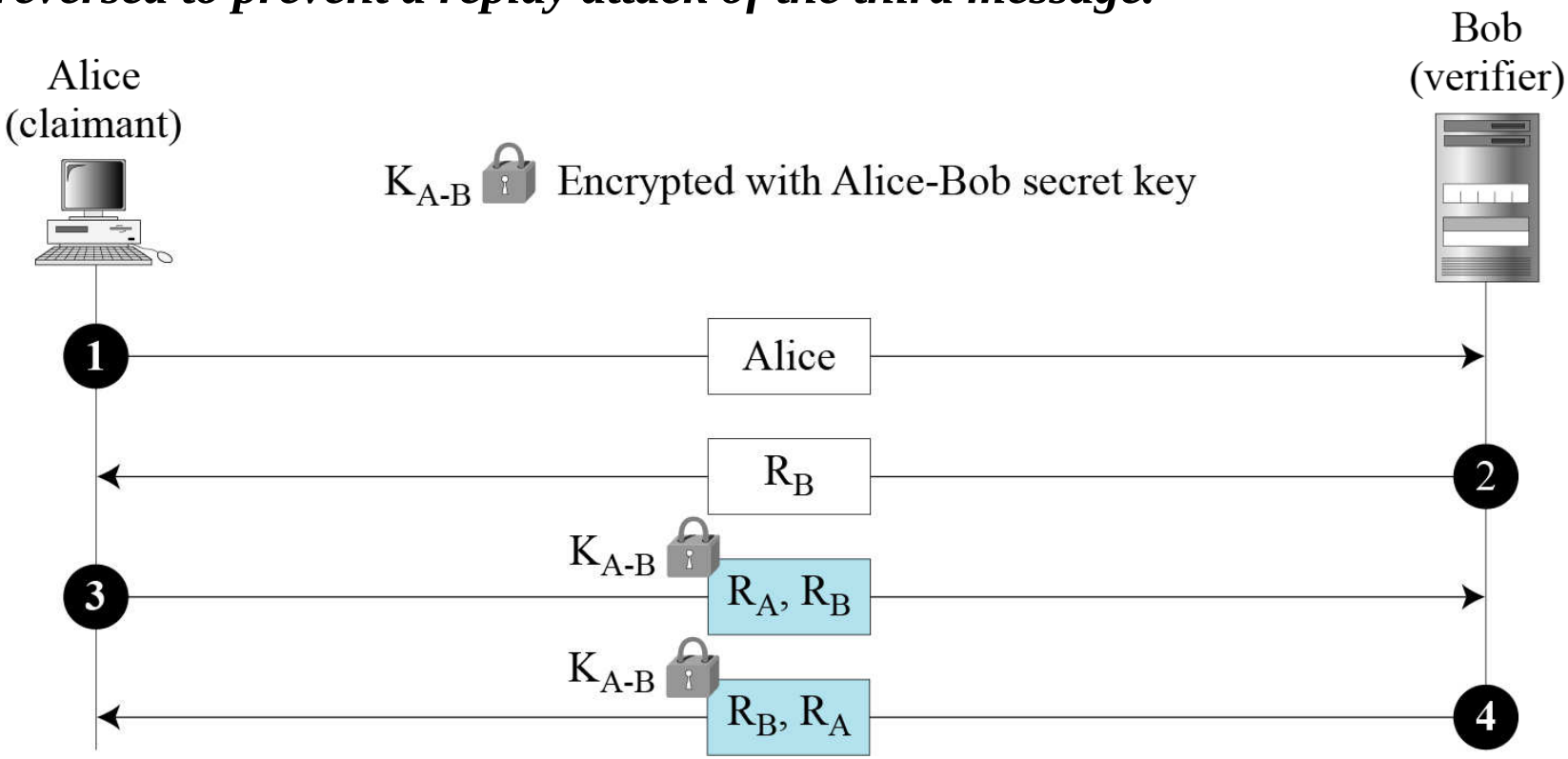
The time-varying challenge here is a timestamp. Assuming all clocks are synchronized, the typical three rounds of message exchange for a nonce challenge are reduced in the case of timestamp challenge to a single message sent by the claimant.

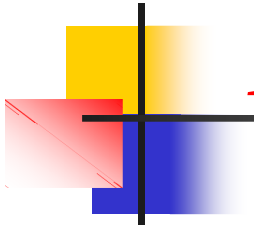


Using a Symmetric-Key Cipher

Bidirectional Authentication with Nonce Challenge

Bidirectional authentication uses a challenge for each way of communication. In the third message, Alice responds to Bob's challenge and at the same time sends her challenge to Bob. In the fourth message, the order of the two nonces is reversed to prevent a replay attack of the third message.





Notation for Message Exchanges

The Forouzan textbook uses diagrams to describe the various message exchange protocols. The Stallings textbook uses a more formal notation to describe these protocols. In CNT 4403, we will use a notation somewhat similar to that of the Stallings textbook.

The notation to be used in describing message exchanges is explained by the following example.

$$A \rightarrow B: \qquad A, B, R_a, E_k(R_b)$$

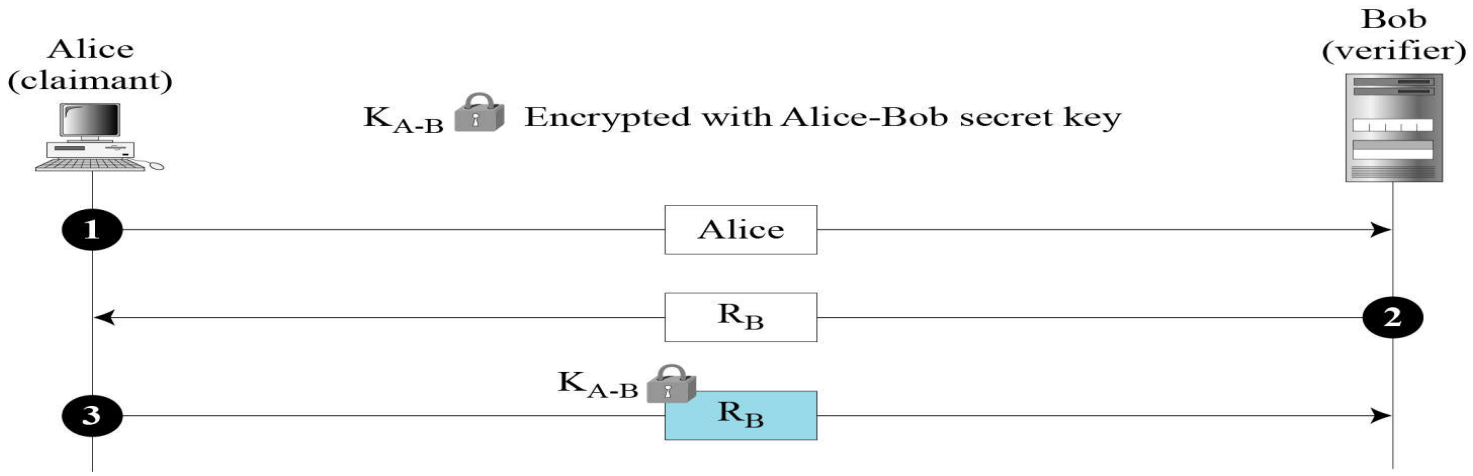
Entity A has sent a message to entity B. The message consists of four values: the user ID of entity A in plaintext, the user ID of entity B in plaintext, the nonce R_a in plaintext and the nonce R_b encrypted using the key K.

Notation for Message Exchanges

Example :

The following protocol is equivalent to the diagram shown below.

$$A \rightarrow B: \quad A$$
$$B \rightarrow A: \quad R_B$$
$$A \rightarrow B: \quad E_{k_{A\text{-}B}}(R_B)$$



Notation for Message Exchanges

Symmetric Protocols:

If the protocol is symmetric, it can be initiated by either one of the two entities. In this case, the variables A and B should be interpreted as generic variables. For example, consider the protocol

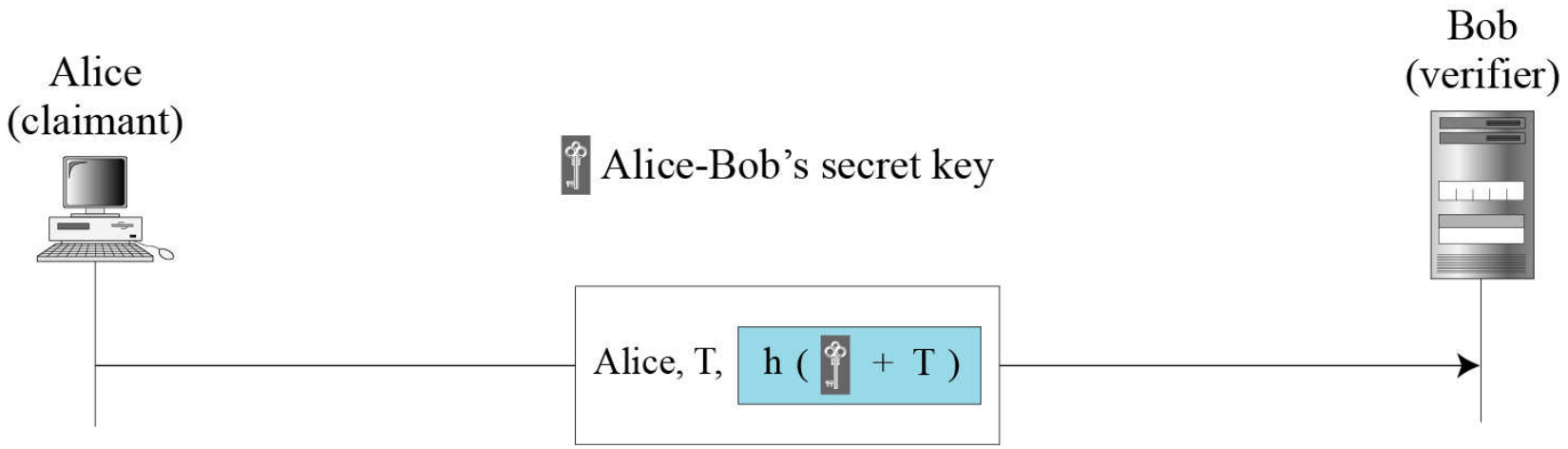
$$A \rightarrow B: \quad A$$
$$B \rightarrow A: \quad R_B$$
$$A \rightarrow B: \quad E_{k_{A-B}}(R_B)$$

If the above protocol is initiated by entity B, it will be equivalent to

$$B \rightarrow A: \quad B$$
$$A \rightarrow B: \quad R_A$$
$$B \rightarrow A: \quad E_{k_{B-A}}(R_A)$$

Using Keyed-Hash Functions

Instead of using encryption/decryption for entity authentication, we can also use a keyed-hash function (MAC). The timestamp is sent both as plaintext and as text scrambled by the keyed-hash function. The receiver takes the plaintext T, applies the keyed-hash function and compares the result with the received hashed value to determine the authenticity of the sender.

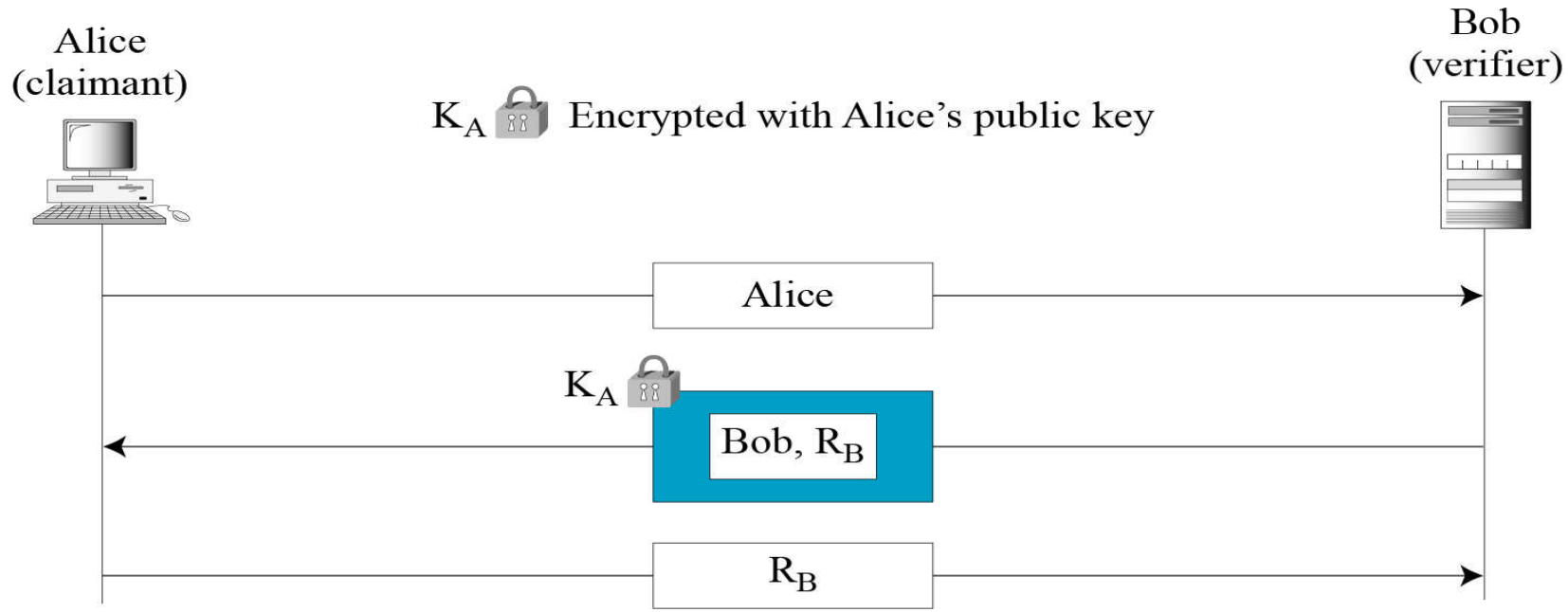


Using an Asymmetric-Key Cipher

The secret used for authentication is the private key of the claimant. The verifier encrypts the challenge using the public key of the claimant. The claimant decrypts the challenge message using his private key and sends the decrypted challenge as the response.

Unidirectional asymmetric-key authentication

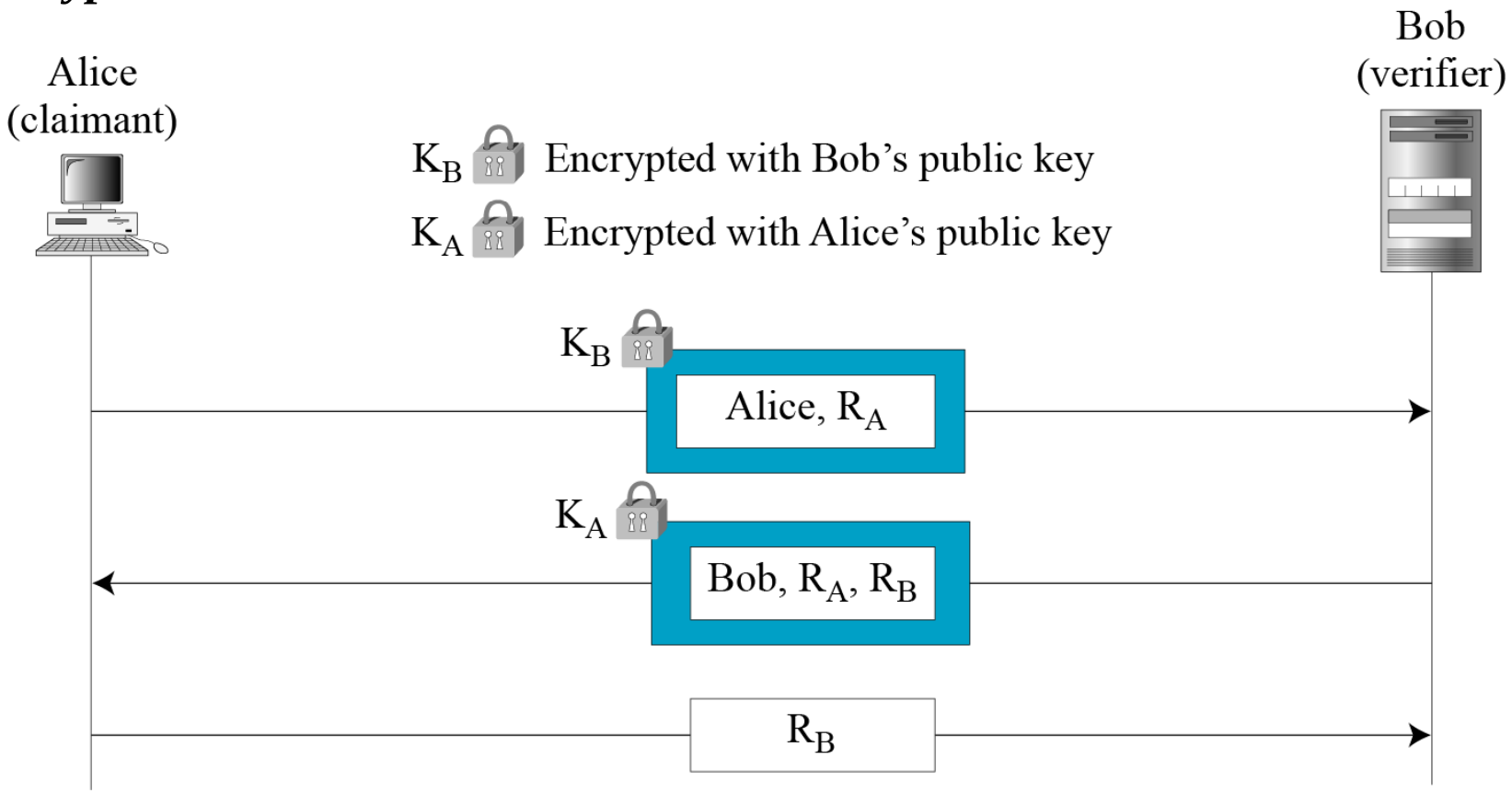
Bob encrypts the challenge using Alice's public key. Alice decrypts the challenge (nonce) with her private key and sends the response (the same nonce) to Bob.



Using an Asymmetric-Key Cipher

Bidirectional asymmetric-key authentication

Two public keys are used, one in each direction. Alice sends her ID and nonce encrypted with Bob's public key. Bob responds with his nonce encrypted with Alice's Public key. Finally, Alice responds with Bob's decrypted nonce.

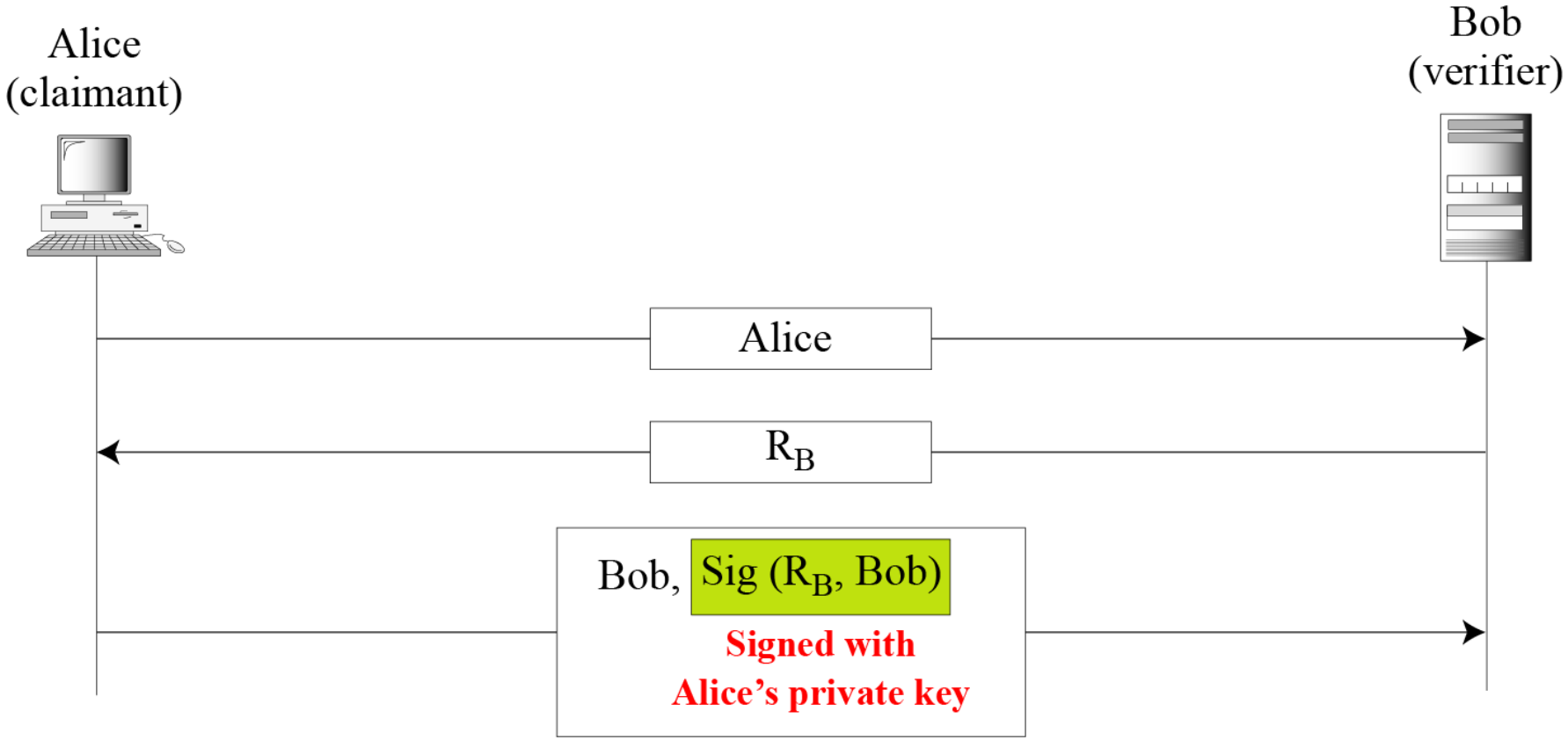


Entity Authentication Using Digital Signature

When digital signature is used for authentication, the claimant uses his private key for signing.

Digital signature, unidirectional Authentication

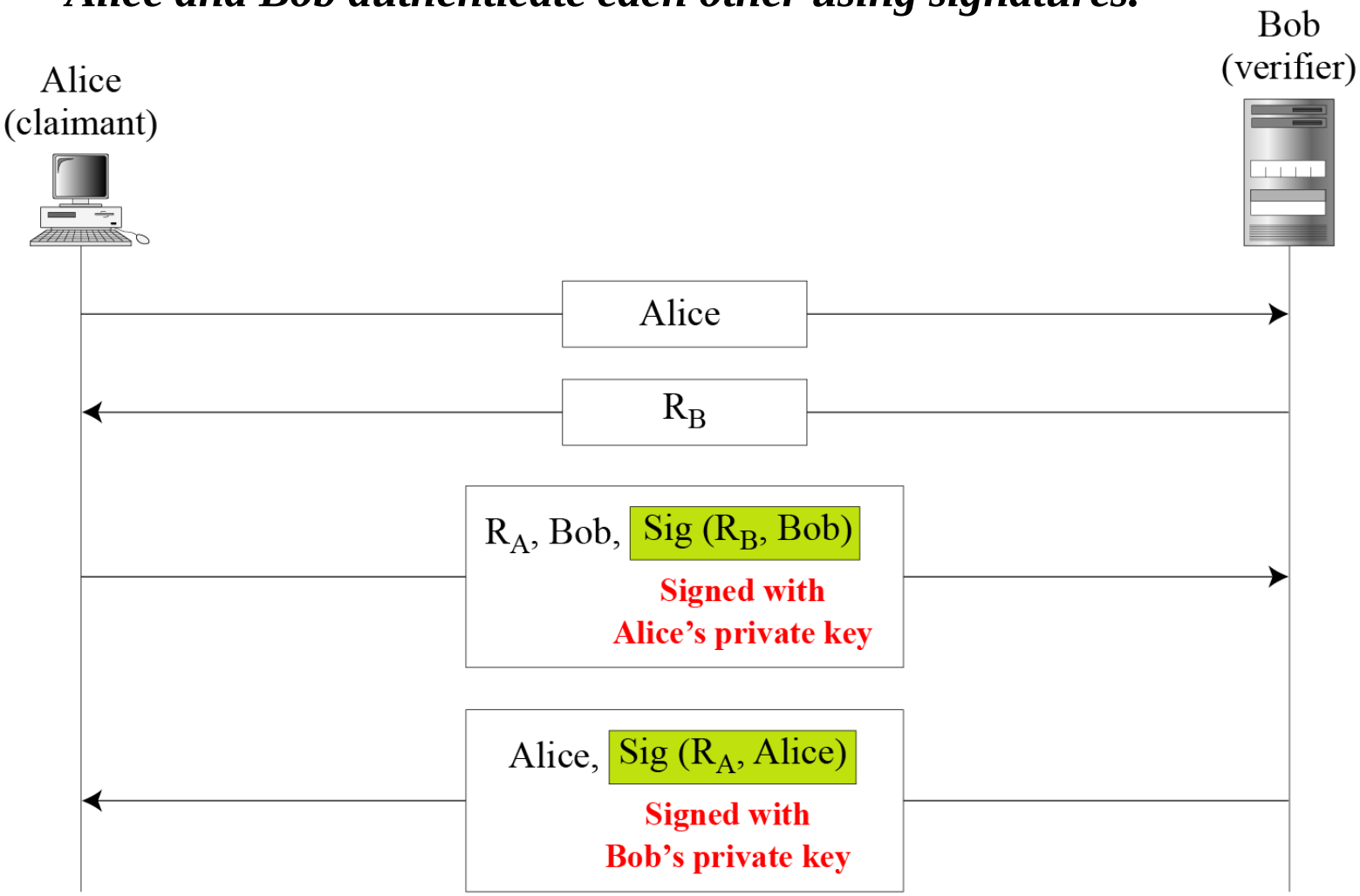
Bob uses a plaintext challenge and Alice signs the response.



Entity Authentication Using Digital Signature

Digital signature, bidirectional authentication

Alice and Bob authenticate each other using signatures.



Zero-knowledge

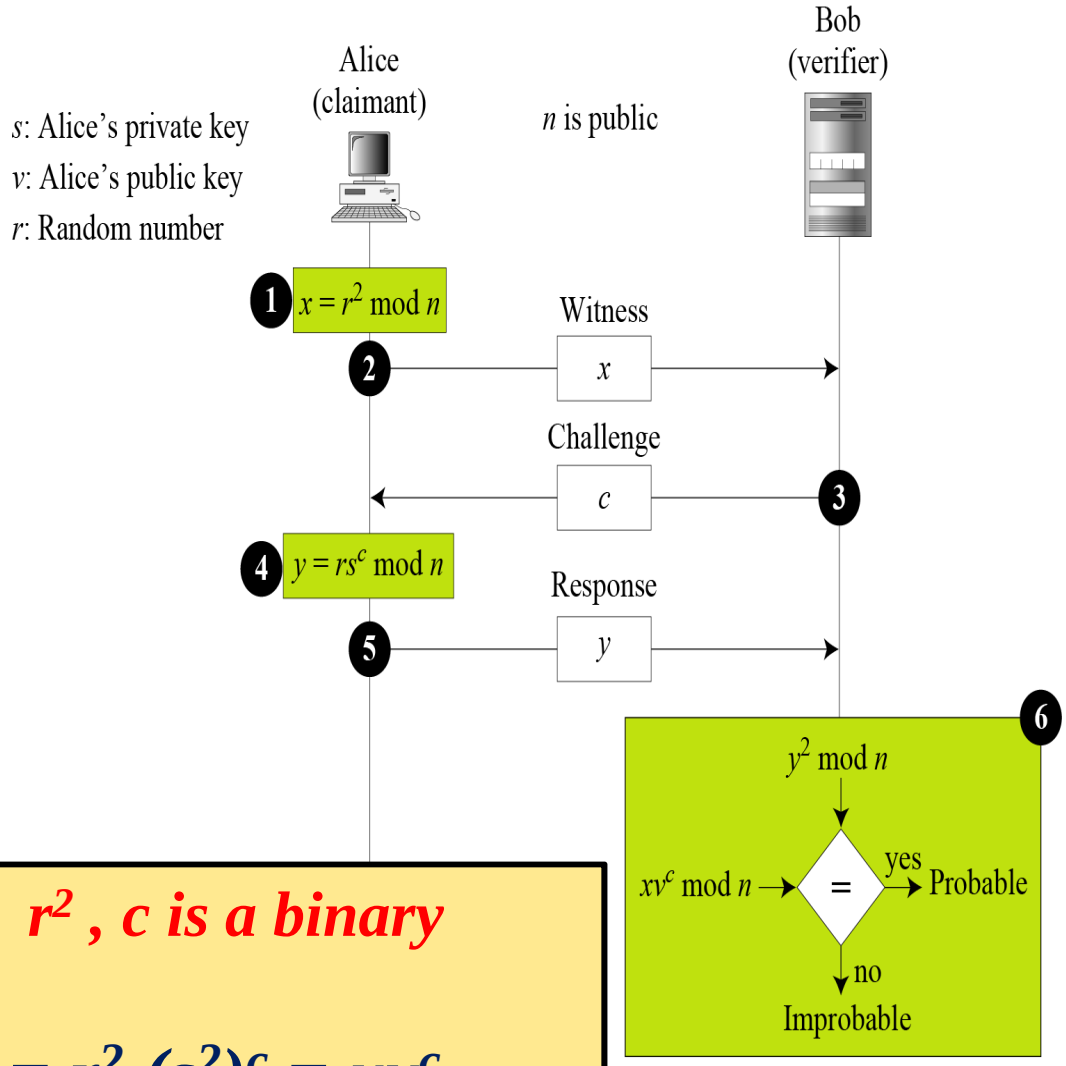
In zero-knowledge authentication, the claimant does not reveal anything that might endanger the confidentiality of the secret. The claimant proves to the verifier that he knows a secret, without revealing it. The interactions are so designed that they cannot lead to revealing or guessing the secret. Zero-knowledge authentication is a more sophisticated type of challenge-response authentication methods.

Fiat-Shamir Protocol

The Fiat-Shamir protocol is based on the difficulty of calculating a square-root modulo large number. The claimant proves knowledge of a secret which is equal to the square root $s = (v)^{\wedge}0.5$ modulo a large modulus n.

In Fiat-Shamir protocol, a trusted third party chooses two large prime numbers p and q to calculate the value $n = p \times q$. The value n is announced to the public. The values of p and q are kept secret. The claimant chooses a secret number s between 2 and n-2 (inclusive). The claimant calculates $v = s^2 \mod n$

The claimant keeps s as the private key and registers v as the public key with the third party.

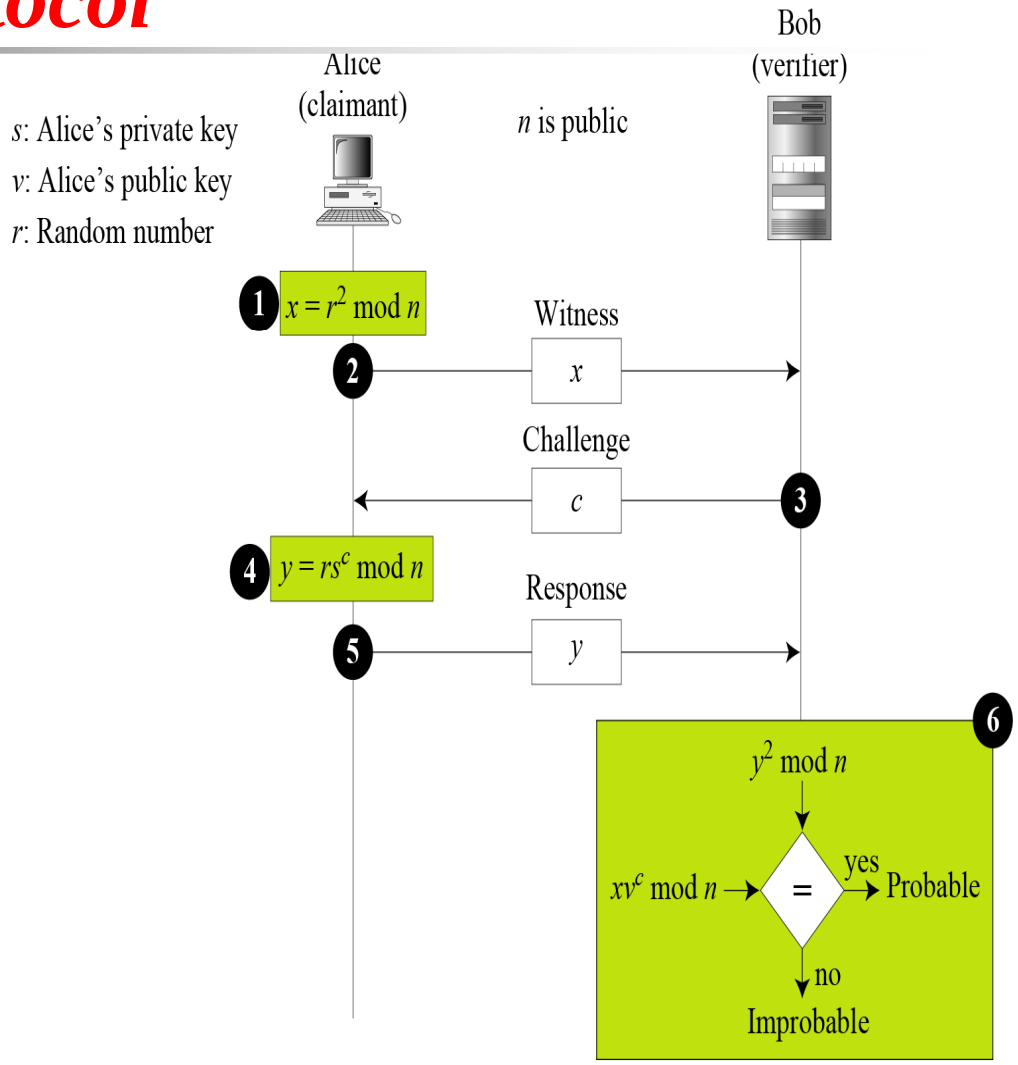


If r is a random number, $x = r^2$, c is a binary value and $y = rs^c$ then

$$y^2 = (rs^c)^2 = r^2 s^{2c} = r^2 (s^2)^c = xv^c$$

Fiat-Shamir Protocol

1. *Alice chooses a random number r between 0 to n-1. She calculates the witness value* $x = r^2 \bmod n$.
2. *Alice sends x to Bob.*
3. *Bob sends the binary challenge c to Alice; c is either 0 or 1.*
4. *Alice computes the response* $y = rs^c \bmod n$ *where s is Alice's private key.*
5. *Alice sends the response y to Bob to prove that she knows Alice's private key.*
6. *Bob calculates* y^2 *and* xv^c. *If they are congruent, then either Alice is honest and knows the private key s or she just made a correct prediction of c.*



Fiat-Shamir Arithmetic Property

$$y^2 = r^2 s^{2c} = xv^c$$

If c=0 ⇒ y^2 = x If c=1 ⇒ y^2 = xv

Fiat-Shamir Protocol

$$x = r^2 \bmod n$$
$$y = rs^c \bmod n$$
$$y^2 = r^2 s^{2c} = xv^c$$

- The response is $y = rs^c \bmod n$
  - If the challenge is $c=0$, the response is $y = r$. This is an easy case for the attacker since the response is simply the random number selected by the attacker who pretends to be Alice.
  - If the challenge is $c=1$, the response is $y = rs$. This is a problem for the attacker since it requires knowledge of s, the private key of Alice.
- To overcome the above problem, the attacker tries to predict the value of c then choose the value of the witness x based on this prediction so that the response y is always equal to the value r. This is done as follows.
  - If the attacker predicts that c =0, he calculates $x = r^2 \bmod n$ which is the correct (honest) way to calculate x. In this case, the response y is equal to r.
  - If the attacker predicts that c =1, he calculates $x = r^2/v \bmod n$ which is dishonest as it is different from the way Alice would have calculated it. In this case, the value of y^2 is equal to $xv^c = xv = (r^2/v)v = r^2$. Thus in this case also, the response y is equal to r.
- If the attacker predicts the wrong value of c, his calculation of the value of the witness x would lead to a wrong response.

Fiat-Shamir Protocol

$$x = r^2 \bmod n$$
$$y = rs^c \bmod n$$
$$y^2 = r^2 s^{2c} = xv^c$$

Summary

- *If the claimant is a dishonest attacker and does not know Alice's private key s, he will execute steps 1 and 2 (for choosing r) then compute the value of x based on his prediction of the value of the challenge c that will be sent by Bob in step 3. This is done as follows*

case of prediction c = 0 → attacker computes $x = r^2 \bmod n$

case of prediction c = 1 → attacker cheats and computes $x = r^2 / v^c \bmod n$

- *The attacker cheats when the prediction of c =1 so that the value of the response y needed in step 5 can be easily computed as* $y = r$ *without using the formula* $(y = rs^c \bmod n)$ *that requires knowledge of the secret key S.*
- *The correctness of the response y depends on the correctness of the prediction of the value of c.*

Fiat-Shamir Protocol

$$x = r^2 \bmod n$$
$$y = rs^c \bmod n$$
$$y^2 = r^2 s^{2c} = xv^c$$

Summary (continued)

- Each round of the Fiat-Shamir authentication consists of six steps that include three message exchanges. If two rounds of Fiat-Shamir authentication are executed, the dishonest claimant has to make two predictions of c (one prediction in each round). This reduces his chances of success to 25%, i.e., probability of $(0.5)^2$.
- If n rounds of Fiat-Shamir authentication are executed, the probability of success for the dishonest claimant becomes $(0.5)^n$. For n =20, the chances of success is approximately 1 in one million.

Guillou-Quisquater (G-Q) Protocol

The G-Q protocol is a modification of the Fiat-Shamir protocol that requires fewer number of rounds.

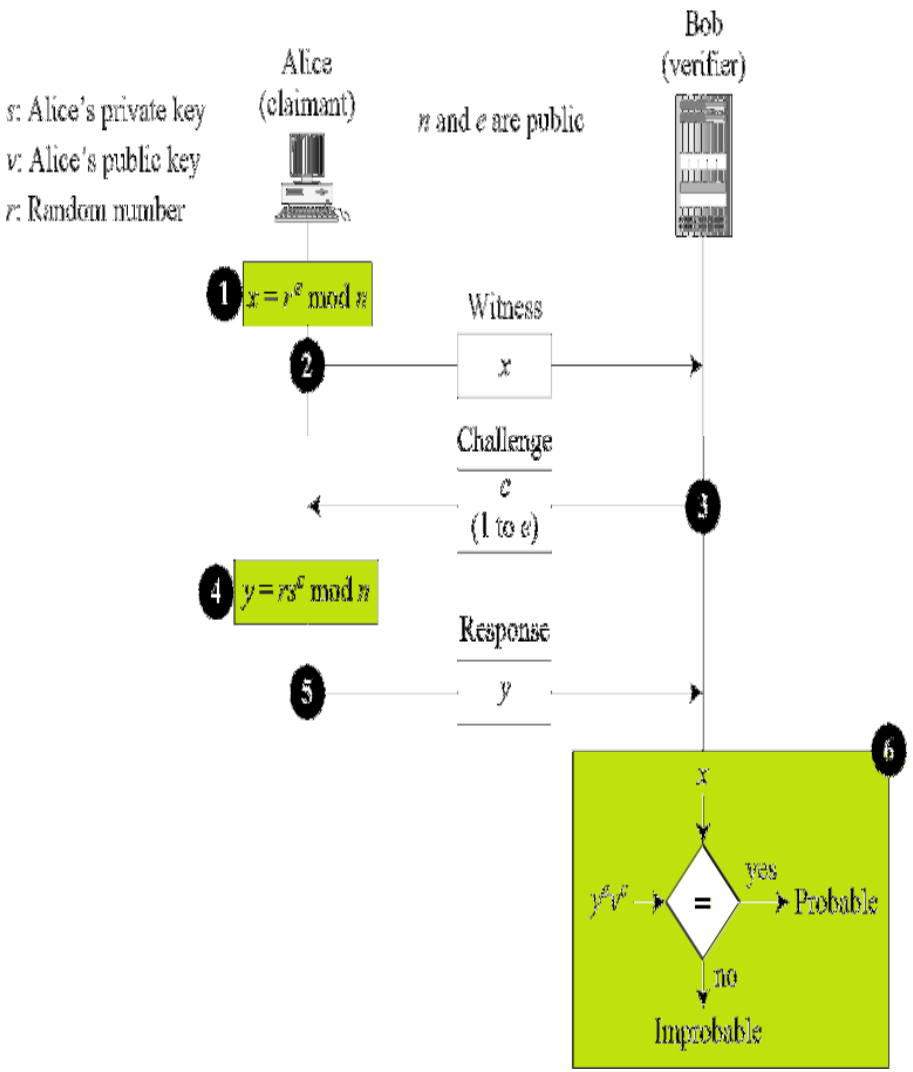
A trusted third party chooses two large prime numbers p and q to calculate the value $n = p \times q$.
The trusted party chooses an exponent e which is coprime with

$$\phi(n)=(p-1)(q-1)$$

The value n and e are announced to the public. The values of p and q are kept secret. The trusted party chooses two numbers for each user: v which is public and s which is secret. The relationship between v and s is

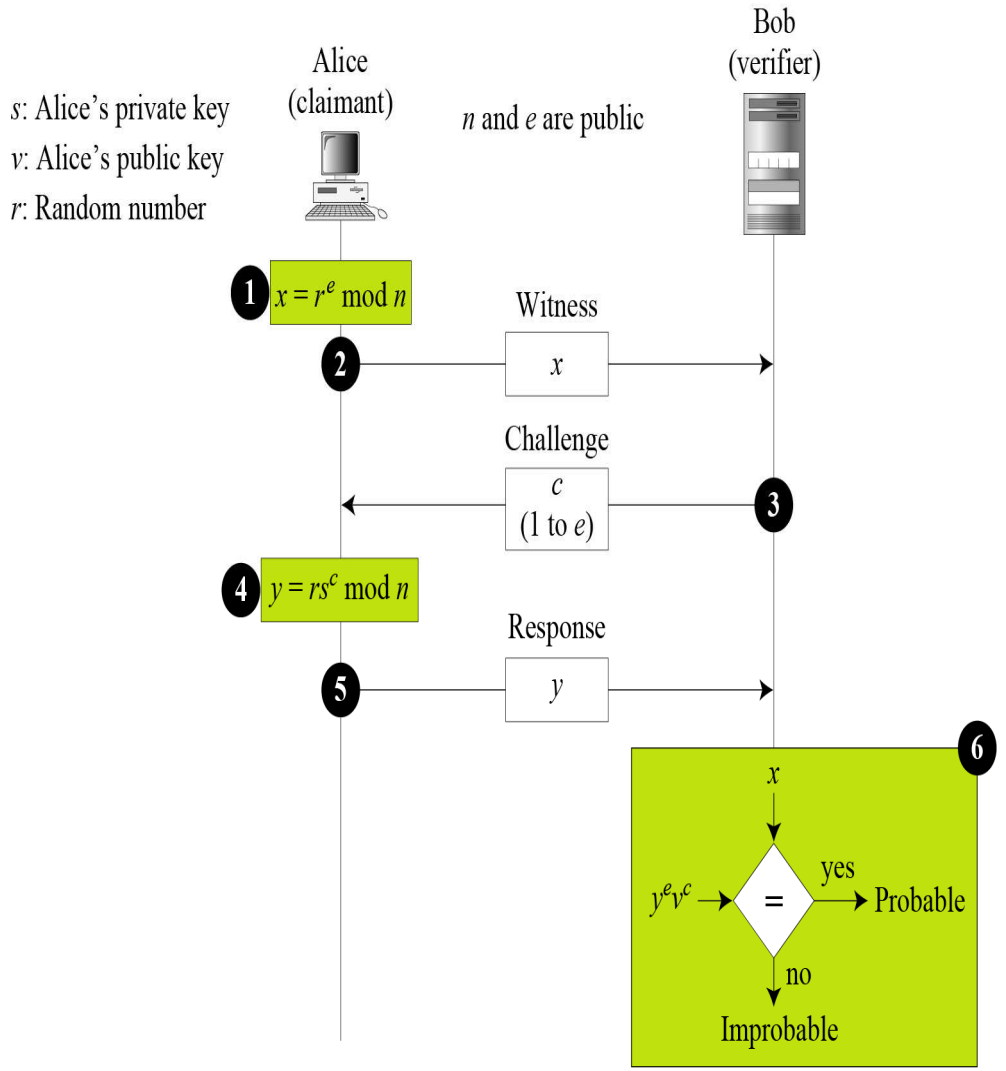
$$v \times s^e \mod n = 1$$

The three message exchanges constitute one round. The value of the random challenge c is between 1 and e.



Guillou-Quisquater (G-Q) Protocol

1. Alice chooses a random number r between 0 to n-1. She calculates the witness value $x = r^e \bmod n$.
2. Alice sends x to Bob.
3. Bob sends the challenge c to Alice; c is between 1 and e.
4. Alice computes the response $y = r\, s^c \bmod n$ where s is Alice's private key.
5. Alice sends the response y to Bob to prove that she knows Alice's private key.
6. Bob calculates $y^e v^c$ And compares it with x. If they are congruent, then Alice is either honest or guessed c correctly.



$$y^e\, v^c = (rs^c)^e\, v^c = r^e\, s^{ce} v^c = r^e\, (s^e\, v)^c = x\, 1^c = x$$

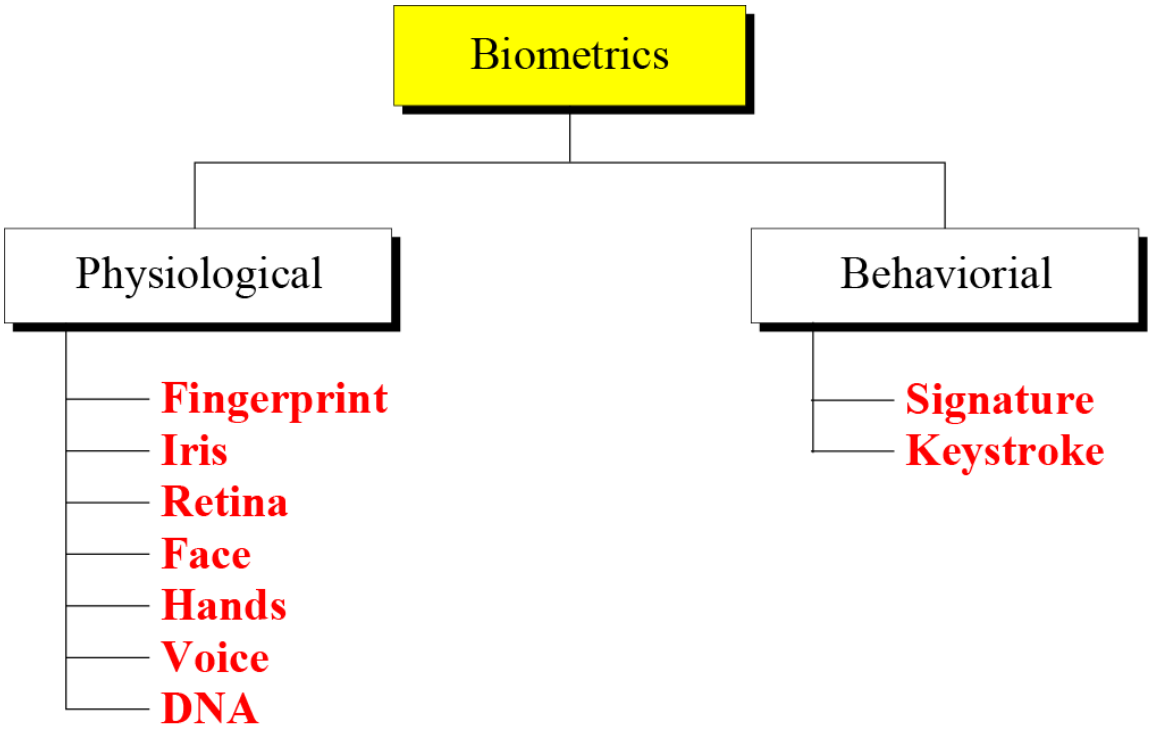
Biometrics

Biometric measurements deal with the physiological or behavioral features that identify a person and that cannot be guessed, stolen, or shared (authentication by something inherent).

Components: ***Several components are needed for biometrics, including capturing devices, processors, and storage devices.***

Enrollment: ***Before using biometric techniques for authentication, the corresponding features of each person in the community should be available in the database. This is referred to as enrollment.***

Biometric Authentication Techniques



Measuring Accuracy of Biometric Techniques

False Rejection Rate (FRR): ***measures how often an authentic person is not recognized by the biometric system. FRR is the ratio of false rejections to the total number of attempts.***

False Acceptance Rate (FAR): ***measures how often a fraudulent person is recognized by the biometric system as a legitimate user. FAR is the ratio of false acceptances to the total number of attempts.***

Applications

Several applications of biometrics are already in use. In commercial environments, these include access to facilities, access to information systems, transaction at point-of-sales, and employee timekeeping. In the law enforcement system, they include investigations (using fingerprints or DNA) and forensic analysis. Border control and immigration control also use some biometric techniques.