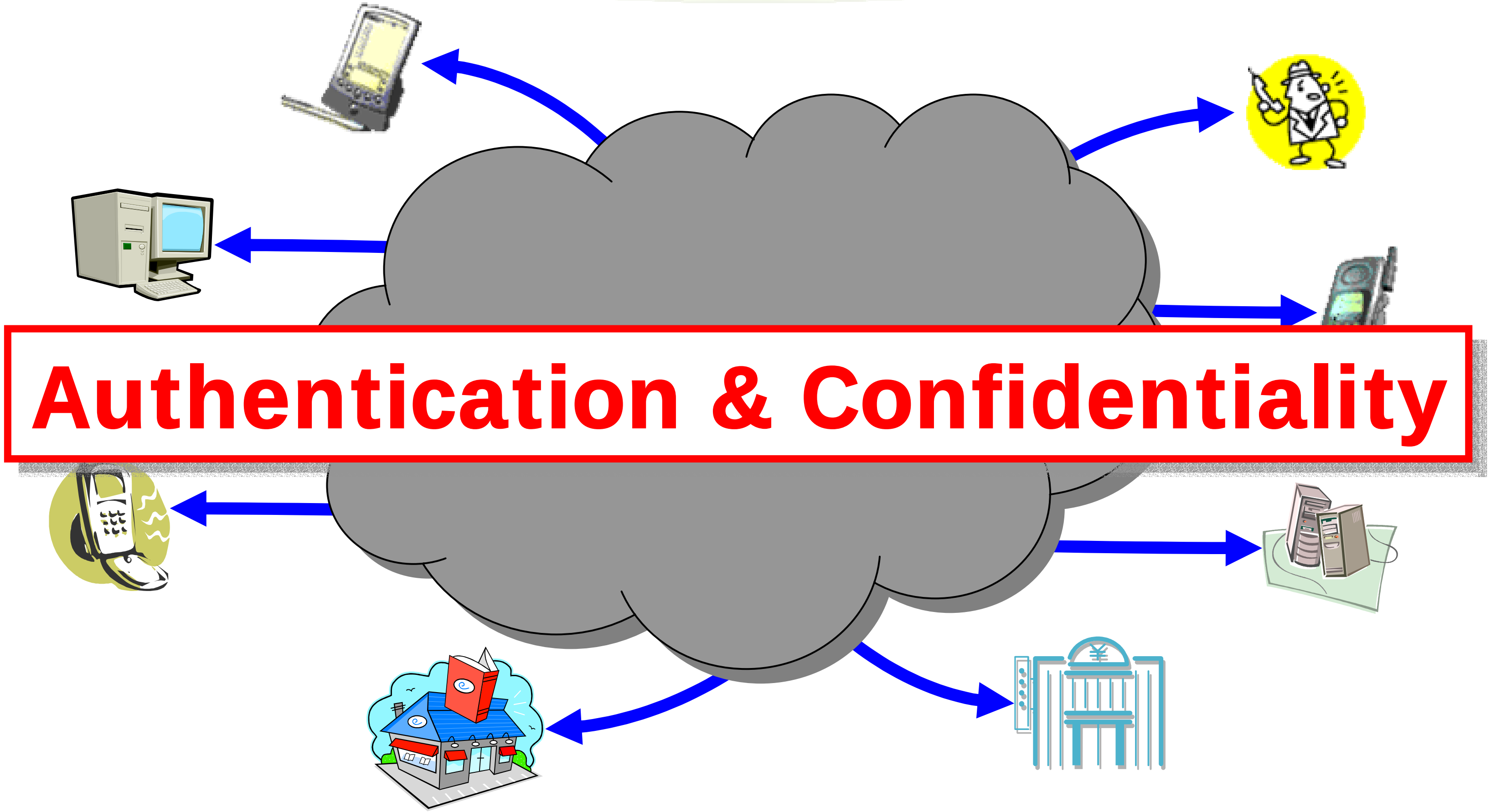
A Lower-Bound of Complexity for RSA-Based Password-Authenticated Key Exchange

SeongHan Shin, Kazukuni Kobara, and Hideki Imai

University of Tokyo, JAPAN

Fundamental Security Goals



Authentication & Confidentiality

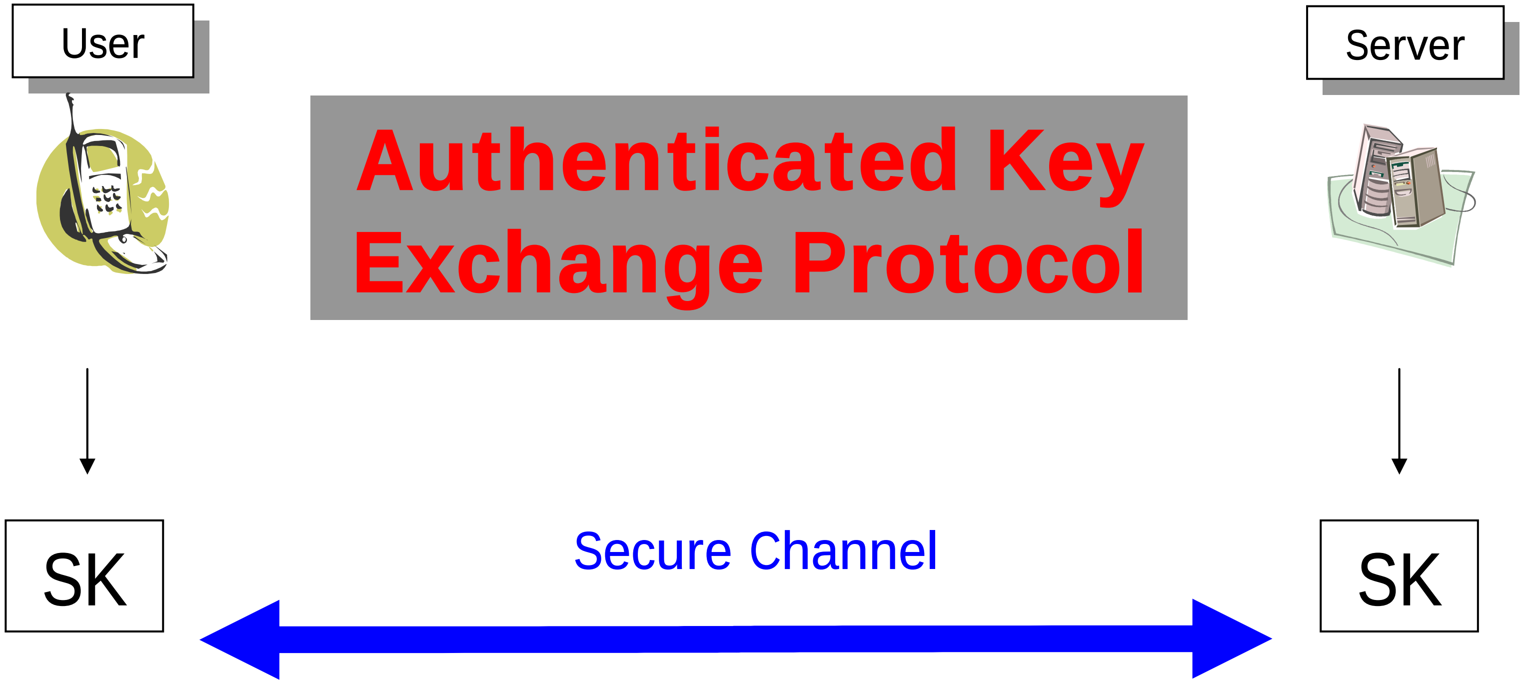
We need something in order to secure the communications.

Authenticated Key Exchange

- Authenticated Key Exchange (AKE) protocols

  both mutual authentication and generation of cryptographically-secure session keys in a secure way

  - A combination of authentication and key exchange



User

Server

Authenticated Key Exchange Protocol

SK

Secure Channel

SK

Classification by Authentication

Which kind of information is needed for authentication

- ❑ AKE based on PKI (Public Key Infrastructures)
  - PKI (WPKI) is required.
  - IKE (Internet Key Exchange), SSL/TLS and SSH
- ❑ AKE based on SK (Strong Secrets)
  - Via symmetric key encryption or message authentication
- ❑ AKE based on PK (Public Keys) and PW (Weak Secrets)
  - No security infrastructures (e.g., PKI)
- ❑ AKE based on PW (Weak Secrets)
  - Neither security infrastructures nor device (for user)

Classification by Key Exchange

Which kind of KE protocol is needed for generating session keys

- ☐ AKE based on KA (Key Agreement) Protocol

  e.g., Diffie-Hellman protocol

  The Diffie-Hellman key is used to compute authenticators and a session key.

- ☐ AKE based on KT (Key Transport) Protocol

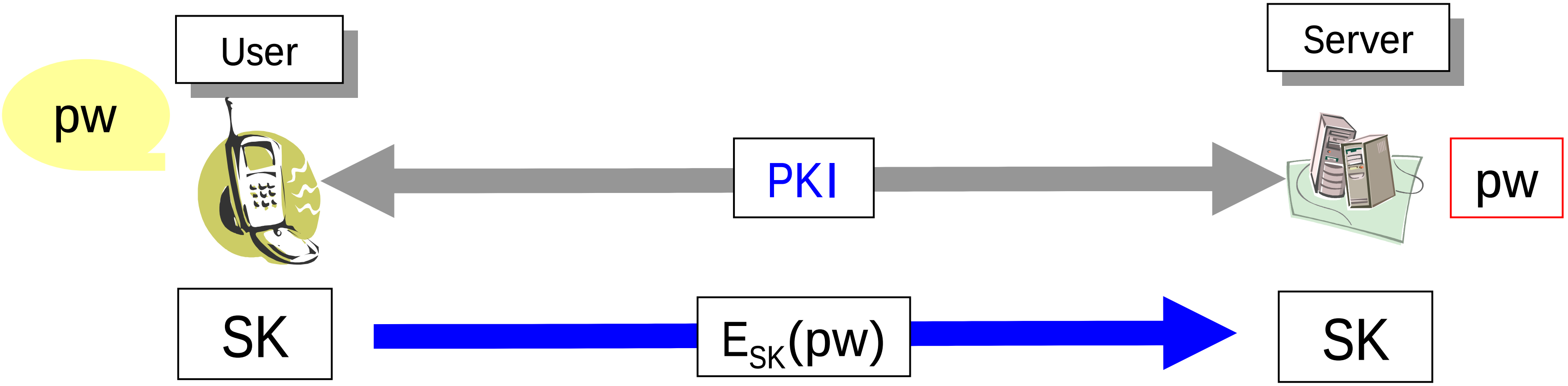
  using symmetric-key (e.g., AES) or public-key encryption (e.g., RSA)

  The KM (keying material) is used to compute authenticators and a session key.

- Authenticators are needed to ensure whether each party has a correct Diffie-Hellman key or keying material or not.

SSL/TLS, SSH

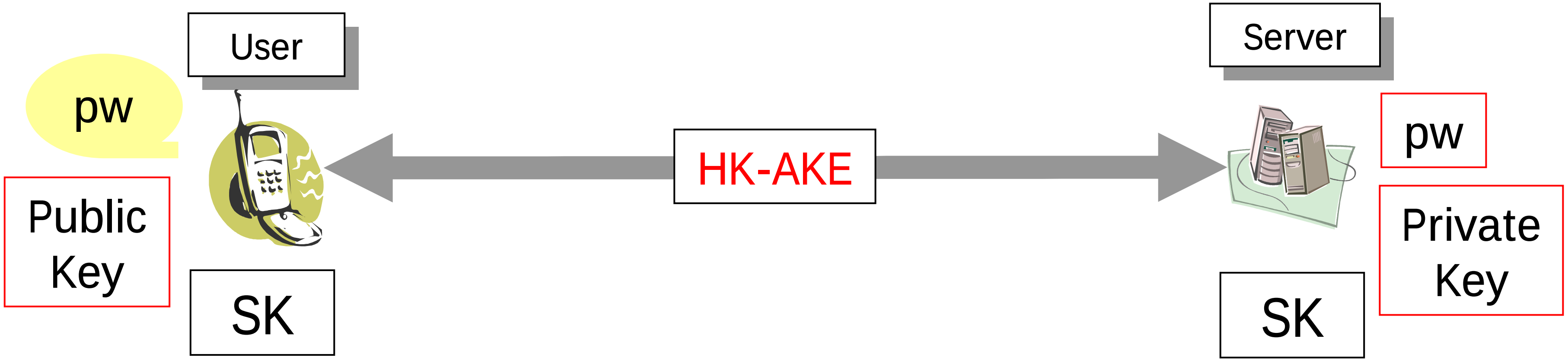
- SSL/TLS, SSH based on (PKI+PW)
  - Password-based user authentication mode



  - Management of public keys and its validity check through CRL (Certificate Revocation Lists) or OCSP (Online Certificate Status Protocol)
  - Burden of PKI

HK-AKE

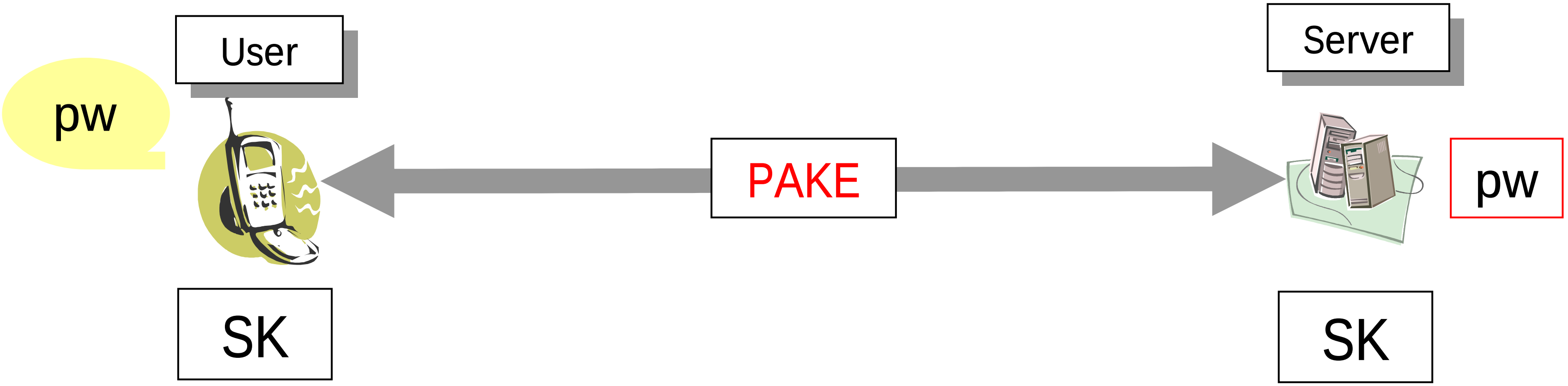
- HK-AKE (Halevi and Krawczyk's AKE [HK99])
  - A user remembers a password and stores a server's public key in advance.



[HK99] S. Halevi and H. Krawczyk, "Public-Key Cryptography and Password Protocols", ACM Transactions of Information and System Security, 1999

PAKE

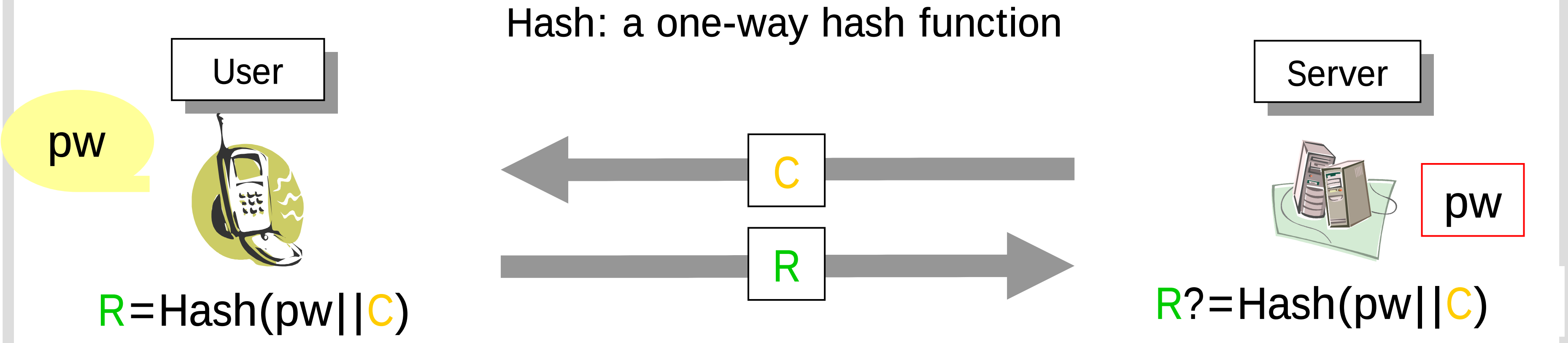
- PAKE (Password-Authenticated Key Exchange)
  - A user remembers only password (without any device).
  - IEEE P1363.2 (in standardization)



  - Only 2-party setting
  - Inefficient in order to verify a server's RSA public key

CHAP

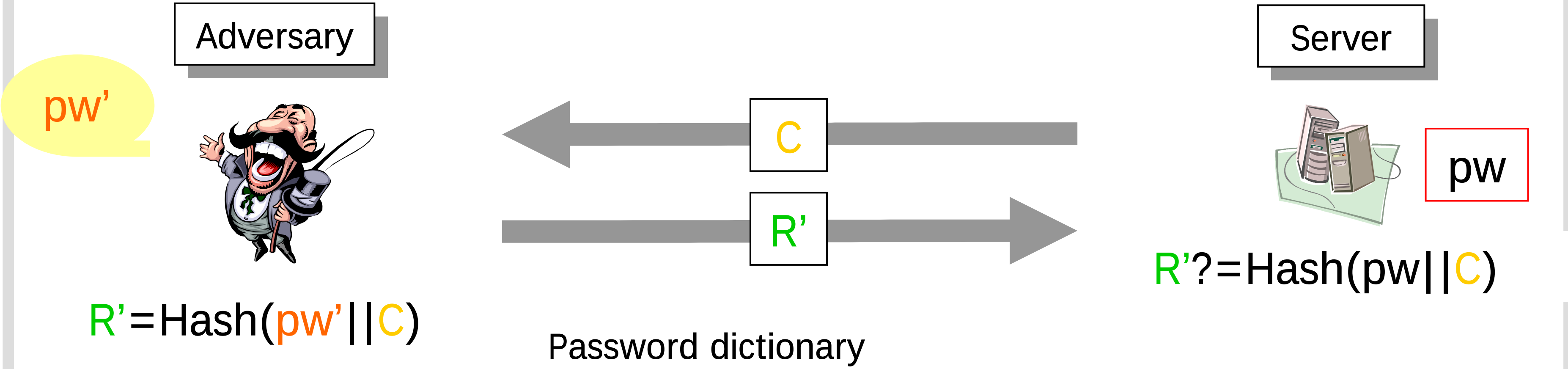
- Challenge-response HAndshake Protocol (CHAP) mainly used in PPP (Point to Point Protocol) for dial-up connection.


Hash: a one-way hash function

- Hash is a *secure one-way* hash function such that
  (i) it is easy to compute Hash(X) and
  (ii) it is hard to compute X from Hash(X).

On-line Attack on CHAP

- On-line dictionary attacks



R'=Hash(pw'||C)

R'?=Hash(pw||C)

Password dictionary

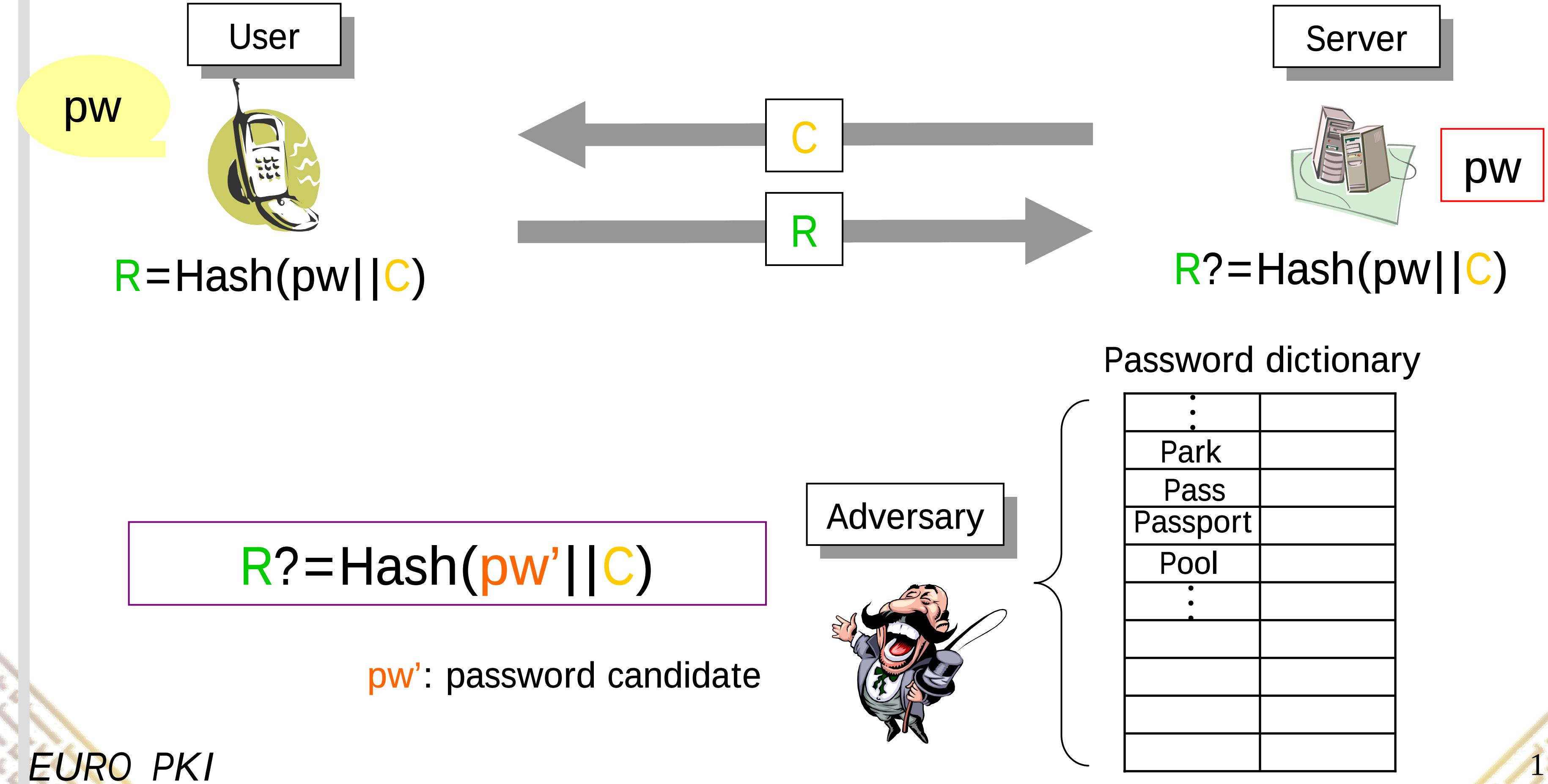
pw': password candidate

⋮	
Park	
Pass	
Passport	
Pool	
⋮	

On-line attacks can be easily prevented by letting a server take appropriate intervals between invalid trials.

Off-line Attack on CHAP

- ❑ Vulnerable to off-line dictionary attacks



User

pw

Server

pw

C

R

R=Hash(pw||C)

R?=Hash(pw||C)

Password dictionary

⋮	
Park	
Pass	
Passport	
Pool	
⋮	

Adversary

R?=Hash(pw'||C)

pw': password candidate

AKE?!

A combination of password-based authentication and RSA-based key transport protocol

- Password-based authentication is a legacy solution.

  usability of passwords and convenience

- The RSA encryption function is fast.

  high efficiency for user's low-power computing devices

  For computing one modular exponentiation, it requires around quadratic running time in the bit-length of its inputs.

  When e=3,

$$RSA_{N,3}(x) \equiv x^3 \bmod N$$

Brief History of PAKE

- Bellovin and Merritt [BM92]

  discussed about the problem of off-line dictionary attacks

  first showed the feasibility that a combination of symmetric and asymmetric (public-key) cryptographic techniques can provide insufficient information for an adversary to verify a guessed password and thus defeat off-line dictionary attacks

  Their paper became the basis for Password-Authenticated Key Exchange (PAKE)

- Until 2000,

  Many password only protocols without provable security

- Up to present,

  Provably-secure and practical (DH or RSA-based) PAKE protocols

Brief History of RSA-based PAKE

- Bellovin and Merritt

  RSA-based Encrypted Key Exchange

  e-residue attacks

  insecure

- Provably-secure RSA-based PAKE

  MacKenzie at Asiacrypt 2000

  – the exponent e should be greater than n

  Catalano at Crypto 2004

  – e can be a small value (e=3 or 2^{16}+1)

  – suitable for the low-power computing devices on client side

  Zhang at Asiacrypt 2004

  – number-theoretic techniques

Interactive Protocol

- e-residue attack

  Adversary can exploit the RSA public key (e,n) s.t. $\gcd(e,\varphi(n))\neq 1$

  The basic idea is that the RSA encryption is no longer a permutation, which maps an element x to the set of e-residues. Since the adversary knows the factorization of n, it is easy to check whether an element is e-residues or not.

- In order to avoid e-residue attack, it is one of the ways to use "interactive protocol".

Motivation and Contribution

- The previous RSA-based PAKE protocols (including Catalano's one) which exploit a challenge-response method for verifying the validity of a RSA public key didn't specify the lower-bound of complexity of their protocols.

- We show a RSA-based PAKE protocol when e is a small number.

- We deduce its lower-bound of complexity along with the actual computation and communication costs.

Notations

(e,n),(d,n): an RSA public/private key pair

RSA: the RSA encryption with (e,n)

G: a full-domain hash (FDH) function $\{0,1\}^* \rightarrow \mathbb{Z}_N{}^* \backslash \{1\}$

H: a one-way hash function

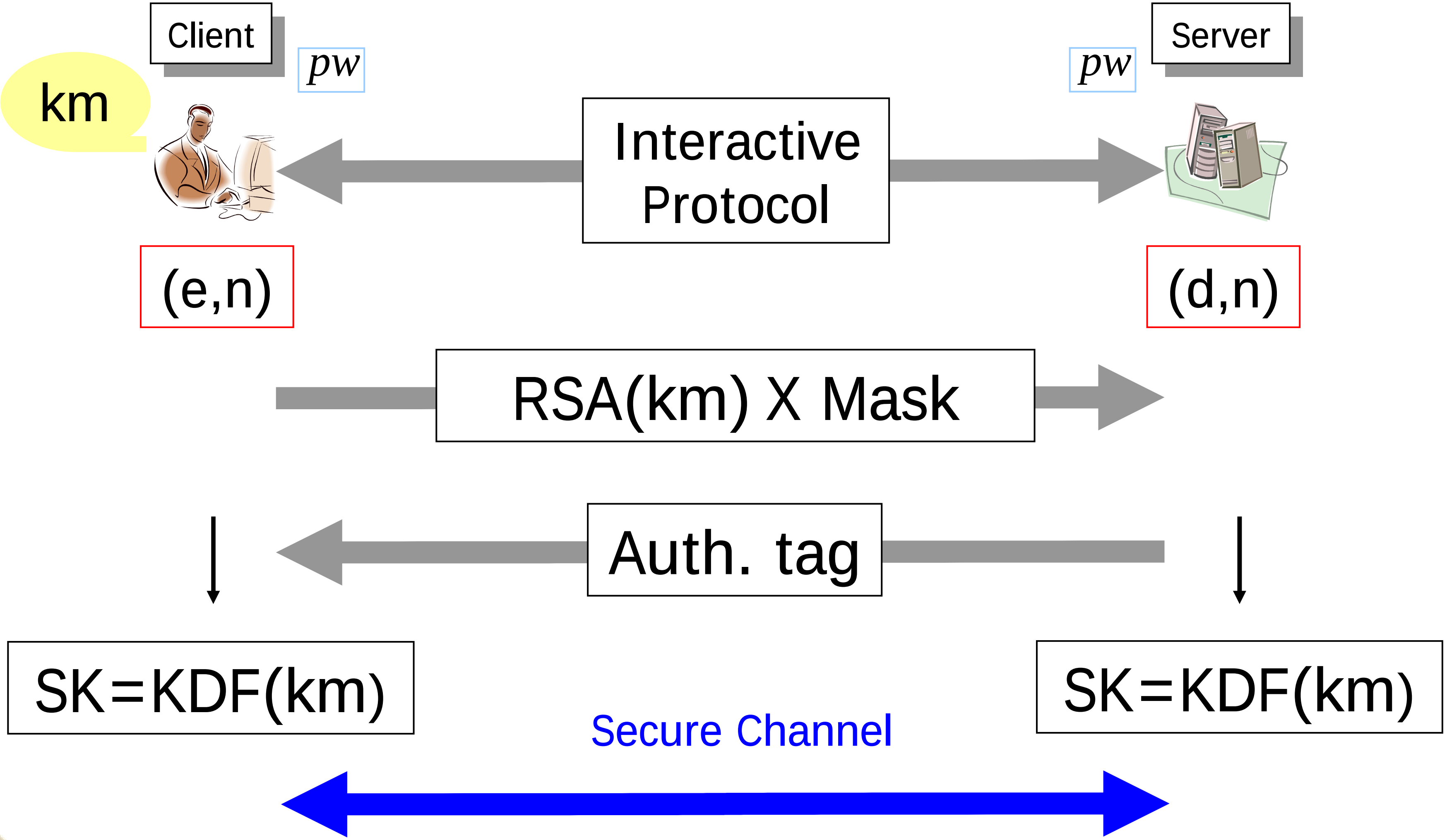
pw: user's password

km: a keying material (e.g., a random number)

Auth: an authenticator

KDF: a key derivation function

Overall Protocol



Client

pw

km

Server

pw

Interactive Protocol

(e,n)

(d,n)

RSA(km) X Mask

Auth. tag

SK=KDF(km)

SK=KDF(km)

Secure Channel

Concrete Construction (1/2)

Client pw

Server pw

(e,d,n) <- RSAKeyGen(1^k)

$\xleftarrow{\quad (e,n) \quad}$

r <- $\{0,1\}^k$

$\xrightarrow{\quad r \quad}$

For i=1 to l

$y_i = H(n,r,i)$, $x_i = y_i^d \bmod n$

$\xleftarrow{\quad \{x_i\} \quad}$

For i=1 to l

$x_i^e \bmod n \mathrel{?=} H(n,r,i)$

Concrete Construction (2/2)

Client pw

Server pw

(e,n)

(d,n)

$t \leftarrow Z_n^*$, $z = t^e \bmod n$

$PW = G(n,pw)$

$z' = z \times PW$

$\xrightarrow{\quad z' \quad}$

$PW = G(n,pw)$

$t = (z' \times PW^{-1})^d$

$\xleftarrow{\quad Auth=H1(C,S,n,z,pw,t) \quad}$

Auth valid?

$sk = H0(C,S,n,z,pw,t)$

$sk = H0(C,S,n,z,pw,t)$

The Complexity depends on “l”

Client

pw

Server

pw

$(e,d,n) <- RSAKeyGen(1^k)$

r <- {0

Communication costs: |n| X l

Computation costs: l modular exp.

For i=1 to l

$y_i = H(n,r,i)$, $x_i = y_i \bmod n$

$\{x_i\}$

For i=1 to l

$x_i^e \bmod n ?= H(n,r,i)$

Security Definitions

Definition 1 (AKE Security) A protocol P is said to be secure if, when adversary A asks q_{se} queries to Send oracle and passwords are chosen from a dictionary of size N, the adversary's adv. in attacking the protocol is bounded by

$$O(q\_\{se\}/N)+\varepsilon(k)$$

for some negligible function $\varepsilon(\cdot)$ in k.

Definition 2 (One-wayness of RSA)

$$\mathrm{Succ}(I) = \Pr[x'=x \mid (e,d,n) \leftarrow \mathrm{RSAKeyGen}(1^k);$$
$$x \leftarrow Z_n^*;\ y=x^e \bmod n;\ x' \leftarrow I(n,e,y)]$$

The RSA function is one-way if Succ(I) is negligible in k.

Security Proof

- Security proof (refer to Catalano’s paper)

Theorem 1 (AKE Security) For any adversary A within a polynomial time t, with less than q_{se} active interactions with the parties and q_{ex} passive eavesdropping, and asking q_{h}, q_{g} and q_{hj} hash queries to H, G and Hj respectively, the advantage of A in attacking the protocol is upper bounded by

$$\mathrm{Adv}^{ake}(A) \leq 24QX\mathrm{Succ}^{ow}(\cdot, \cdot) + 4QX\boxed{\mathrm{Succ}^{forge}(t)} + 24Q/N + \varepsilon(k)$$

where k is the security parameter and Q ≤ q_{se}+q_{ex}.

e-residue Attack

- Adversary A uses a RSA function that is not a permutation.
  - With the view of z, the adversary tries all the passwords, and only a strict fraction leads to z in the image of RSA enc.
  - But for that, the adversary has to forge a proof of validity for RSA enc.

- Fact 1. For odd integer n and e ($e \geq 3$) such that $\gcd(e, \phi(n)) \neq 1$, any e-residue modulo n should have at least three e-th roots.

- Corollary 1. $\Pr[\text{forge}] \leq (1/3)^l$

How many x_i is required? (1/2)

- The two cases for an adversary to break the protocol
- The first case (on-line attack): the adv. generates the right RSA key pair and then performs on-line exhaustive search attacks.
- The second case (e-residue attack): the adv. deliberately generates the RSA key pair (e,n), such that $e|\phi(n)$, by which off-line exhaustive search attacks are performed.

How many x_i is required? (2/2)

- Theorem 2. For any odd integer e ($e \geq 3$), the lower-bound of l is $\lceil -\log_3(1-(1-j/N)^{1/j}) \rceil$.
  We restrict the success prob. of off-line attack by that of on-line attack.

  $$E(\text{on-line}) \leq E(\text{off-line})$$

  As for each instance j, the expectation value of possible password candidates in off-line attack should be more than or equal to the counterpart in on-line attack.

- However, we should claim that the other terms in the security result are irrelevant to both on-line and off-line attacks.

Efficiency

- When e=3 and N (2^{37}) for alphanumerical passwords with 6 characters, l=24.
- Computation costs on client: (l+1) modular exponentiations τ_{exp} with the exponent e and one modular multiplication τ_{mul}

  51 X τ_{mul} ($\tau_{exp} \approx 2 \cdot \tau_{mul}$)
- Communication costs: (l+3)k + |H1| bits

  27.15625 KB

Conclusion

- We showed a RSA-based PAKE protocol when e is a small number.
- We deduced its lower-bound of complexity along with the actual computation and communication costs.