New Approaches to Password Authenticated Key Exchange based on RSA

Muxiang Zhang
Verizon Communications, Inc.
December 7, 2004

Historical Background

- In 1992, Bellovin and Merritt invented EKE
  - Defeats off-line dictionary attack
  - Works well with Diffie-Hellman
  - Subtleties in RSA-based scheme
- In 1997, Lucks proposed OKE based on RSA
  - Broken in 2000
  - In Asiacrypt'00, MackKenzie et al. invented SNAPI
    - requires special RSA public key (n, e), **prime e > n**
- In ISC'02, Zhu et al. improved Bellovin-Merritt interactive protocol to validate RSA public key
  - Large communication overhead between entities
  - In ISC'03, Bao showed weakness of Zhu et al.'s protocol
  - In ISC'04, Zhang developed powerful attack on Zhu et al.'s protocol
  - In Crypto 04, Catalano et al. presented provably secure version of the interactive protocol

EKE based on Diffie-Hellman

A 

prime number p
generator g of Z^*_p

 B

$0 < a < p$

$E_w(g^a)$ →

$0 < b < p$

← $E_w(g^b), H_1(g^{ab})$

$H_2(g^{ab})$ →

EKE does not require digital certificate!

RSA-based EKE

A 

(n, e, d)

$H^*: \{0,1\}^* \rightarrow Z^*_n$,
$H_1, H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$

 B

$r_A \in \{0,1\}^k$

n, e, r_A →

$a \in Z^*_n$, $r_B \in \{0,1\}^k$
$\lambda = H^*(w, r_A, r_B, A, B)$
$z = \lambda a^e \bmod n$

← z, r_B

$b = (\lambda^{-1} z)^d \bmod n$

$\mu = H_1(b, r_A, r_B, A, B)$ →

$\mu = H_1(a, r_A, r_B, A, B)$?

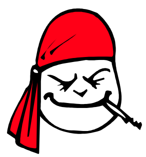
← $\eta = H_2(a, r_A, r_B, A, B)$

$\eta = H_2(b, r_A, r_B, A, B)$?

How does B know (n, e) is a valid RSA public key?

e-Residue Attack

generate: n, e
gcd(e, φ(n)) $\neq$ 1



B

$r_E \in \{0,1\}^k$

n, e, r_E →

$a \in Z^*_n$, $r_B \in \{0,1\}^k$
$\lambda = H^*(w, r_E, r_B, A, B)$
$z = \lambda a^e \bmod n$

← z, r_B

Abort execution

1) Eve selects a random password $\alpha \in D$
2) Eve computes $\lambda = H^*(\alpha, r_E, r_B, A, B)$
3) Eve tests if equation $\boxed{x^e = \lambda^{-1} z \mod n}$ has solution
4) If there is no solution, Eve excludes α from D and returns to step 1; otherwise, Eve returns to step 1.

SNAPI Protocol

RSA: n, e, d
prime e > n > $\varphi(n)$

B

$r_A \in \{0,1\}^k$

$\xrightarrow{\quad n,\ e,\ r_A \quad}$

Is e prime > n ?

$a \in Z^*_n,\ r_B \in \{0,1\}^k$

$\lambda = H^*(w, r_A, r_B, A, B)$

$z = \lambda\, a^e \bmod n$

$\xleftarrow{\quad z,\ r_B \quad}$

$b = (\lambda^{-1} z)^d \bmod n$

$\xrightarrow{\quad \mu = H_1(b, r_A, r_B, A, B) \quad}$

$\mu = H_1(a, r_A, r_B, A, B)$?

$\xleftarrow{\quad \eta = H_2(a, r_A, r_B, A, B) \quad}$

$\eta = H_2(b, r_A, r_B, A, B)$?

Computationally prohibitive for primality test of $e > 2^{1024}$.

Interactive Validation Protocol

A 

(n, e, d)

 B

$r_A \in \{0,1\}^k$

$\xrightarrow{\;n,\ e,\ r_A\;}$

$m_i \in Z_n,\ 1 \le i \le t$

$c_i = m_i^{\,e} \bmod n$

$\xleftarrow{\;c_1,\ c_2,\ \dots\ c_t\;}$

$m'_i = c_i^{\,d} \bmod n$

$b_i = H(m_i')$

$\xleftarrow{\;b_1,\ b_2,\ \dots\ b_t\;}$

$b_i = H(m_i)?\ \ 1 \le i \le t$

Large communication overhead.

Password Enabled Key Exchange Protocol (PEKEP)

- Alice (A) and Bob (B) only shares a password
- Alice can select both large and small primes for e
- Bob does not verify if $\gcd(e, \varphi(n)) = 1$
- Bob does not have to test primality for large e
- Low communication overhead; each flow involves at most one RSA message

Description of PEKEP

Alice (A)

$r_A \in \{0,1\}^k$

$\xrightarrow{r_A,\ n,\ e,\ A}$

Bob (B)

e odd prime? n odd?
If yes, $m = \lfloor \log_e n \rfloor$
$a \in Z^*_n$, $r_B \in \{0,1\}^k$
$\alpha = H(w, r_A, r_B, A, B, n, e)$
If $\gcd(\alpha, n) = 1$, $\lambda = \alpha$
else $\lambda \in Z^*_n$

$z = (\lambda a^e)^{e^m} \bmod n$

$\xleftarrow{r_B,\ z}$

$\alpha = H(w, r_A, r_B, A, B, n, e)$
If $\gcd(\alpha, n) > 1$, $b \in Z_n$
else recover a, b=a

$\xrightarrow{\mu = H_1(b,\ r_A, r_B, A, B, n, e)}$

$\mu = H_1(a, r_A, r_B, A, B, n, e)$?
$sk = H_3(a, r_A, r_B, A, B, n, e)$

$\xleftarrow{\eta = H_2(a,\ r_A, r_B, A, B, n, e)}$

$\eta = H_2(b, r_A, r_B, A, B, n, e)$?
$sk = H_3(b, r_A, r_B, A, B, n, e)$

Security against e-Residue Attack

e odd prime, n odd
$\gcd(e, \varphi(n))>1$

Eve

$r_E \in \{0,1\}^k$

r_E, n, e, A →

Bob

$m = \lfloor \log_e n \rfloor$
$a \in Z^*_n, r_B \in \{0,1\}^k$
$\alpha = H(w, r_A, r_B, A, B, n, e)$
If $\gcd(\alpha, n)=1$, $\lambda = \alpha$
else $\lambda \in Z^*_n$

$z = (\lambda a^e)^{e^m} \bmod n$

← r_B, z

Abort execution

Select a password $\pi \in D$
$\alpha = H(\pi, r_E, r_B, A, B, n, e)$
If $\gcd(\alpha, n) = 1$, but

$(\alpha x^e)^{e^m} = z \bmod n$

has no solution in Z_n, exclude π

Security against e-Residue Attack (continue)

Theorem 1. Let $n = p_1^{a(1)} p_2^{a(2)} \ldots p_r^{a(r)}$ be an odd integer. Let m be a non-negative integer and e an odd prime, such that

$$e^{m+1} \text{ not divides } \varphi(p_i^{a(i)}), \qquad 1 \le i \le r.$$

If z is e^m-th power residue of n, then for any $\lambda \in Z^*_n$, equation

$$(\lambda x^e)^{e^m} = z \mod n$$

always has solution in Z^*_n.

In PEKEP, $m = \lfloor \log_e n \rfloor$, $e^{m+1} > n \ge p_i^{a(i)}$. The condition of Theorem 1 is satisfied, so that Eve can not exclude password π!

Computational Overhead

- Computation time for Alice is $O((\log_2 n)^3)$.
- When e is a small prime, e.g. e =13, computational load on Bob is dominated by m+1 RSA encryptions, with computation time $O((\log_2 n)^3)$.
- When e is large, Bob can replace e by smaller prime c

Alice (A)		**Bob (B)**
$r_A \in \{0,1\}^k$	r_A, n, e, A →	**Select odd prime c** $m = \lfloor \log_c n \rfloor$ $a \in Z^*_n$, $r_B \in \{0,1\}^k$ $\alpha = H(w, r_A, r_B, A, B, n, c)$ If $\gcd(\alpha, n) = 1$, $\lambda = \alpha$ else $\lambda \in Z^*_n$
	← c, r_B, z	$z = (\lambda a^c)^{c^m} \bmod n$

Computationally Efficient Key Exchange Protocol (CEKEP)

- Mitigates computational burden on Bob
- Adds two flows to PEKEP
- Alice and Bob shares a password, Bob selects a small number ε, e.g., $0 < \varepsilon \le 2^{-80}$
- Number of RSA encryptions by Bob turns out to $m = \lceil \log_e \varepsilon^{-1} \rceil < \lfloor \log_e n \rfloor$ (required in PEKEP).
- Computation time for Bob: $O((\log_e \varepsilon^{-1})(\log_2 n)^2)$
  - When $\varepsilon = 2^{-80}$, two-three times faster than DH-EKE.

Description of CEKEP

Alice (A)

Bob (B)

$0 < \varepsilon \leq 2^{-80}$

$r_A \in \{0,1\}^k$

$\xrightarrow{\rho,\ r_A,\ n,\ e,\ A}$

e odd prime? n odd?

If yes, $m = \lceil \log_e \varepsilon^{-1} \rceil$

$\tau \in \{0,1\}^k$

$H(n,e,\rho,\tau,A,B,m) \in Z^*_n$

$\xleftarrow{\tau,\ m}$

$\gamma = H(n,e,\rho,\tau,A,B,m)$

$u = D^m(\gamma)$

$\xrightarrow{u}$

$\gamma = E^m(u)$?

If not, reject.

$\longleftrightarrow$ $\boxed{\text{PEKEP} \quad m = \lceil \log_e \varepsilon^{-1} \rceil}$ $\longleftrightarrow$

Security against e-Residue Attack (for CEKEP)

Theorem 2. Let $n = p_1^{a(1)} p_2^{a(2)} \ldots p_r^{a(r)}$ be an odd integer. Let m be a non-negative integer and e an odd prime. If there exists a prime power $p_i^{a(i)}$, such that

$$e^m \text{ divides } \varphi(p_i^{a(i)}),$$

Then for random integer $\gamma \in Z^*_n$,

$$\Pr(\gamma = u^{e^m} \bmod n) \le e^{-m} \le \varepsilon$$

Formal Security Analysis

- Adversarial Model [BPR00]
  - Send
  - Execute
  - Reveal
  - Test
  - Oracle call
- Definition of security [GL03]
  - Execute(A, i, B, j) $\Rightarrow$ partner (Π^i_A, Π^j_B)
  - Adv $\leq Q_{send}/|D| + neg(k, l)$

Assumptions and Results

- Random Oracles: H, H_1, H_2, H_3
- RSA Assumption

$$\mathrm{Adv}_A^{\mathrm{rsa}}(t) = \mathit{Pr}(x^e{=}c \bmod n: (e, d, n) \leftarrow \mathit{GE}(1^l), c \in \{0, 1\}^l, x \leftarrow A(1^l, c, e, n))$$

Theorem 3. For polynomial-time adversary *A* making Q_{send} queries of type *send*,

$$\mathrm{Adv}_A^{\mathrm{PEKEP}} \leq Q_{send}/|D| + (Q_{execute}+3Q_{send})\mathrm{Adv}^{\mathrm{rsa}}(O(t)) + O((Q_{execute}+2Q_{send})Q_{oh}/2^k)$$

Proof available in http://eprint.iacr.org/2004/033.

Conclusion

- Efficient and secure password-authenticated key exchange protocols can be constructed using RSA
- PEKEP and CEKEP do not restrict the size of RSA public key
- PEKEP and CEKEP do not require public parameters
  - truly "password-only" protocols
- Provable security under RSA assumption and random oracle model