• EKE: Encrypted Key Exchange
  
  − Motivation: Users choose bad passwords, which are often vulnerable to dictionary attacks.
  
  − An attacker simply encrypts a large list of such passwords as the actual passwords are encrypted, and checks for matches.
    * Names and nicknames of family and friends: 50%
    * Sports, TV, and Pop Stars: 30%
    * “Self Obsessed” (sex, stud, goddess, etc): 11%
    * Secure Passwords: only 9%
  
  − Basic Ideas
    * Use a weak password as an encryption key for a session key.
    * Make an attacker devote significant resources to each password guess.
  
  − An implementation using Diffie-Hellman Key Exchange

0. Alice and Bob share knowledge of $\alpha$ and $\beta$, the Diffie-Hellman base and modulus, respectively, and $P$, an encryption key deterministically generated from Alice’s (potentially weak) password.

1. Alice selects a random $R_A$ and transmits (“Alice”, $\{\alpha^{R_A} \mod \beta\}_P$) to Bob.

2. Bob selects a random $R_B$ and a challenge $c_B$, and calculates the session key $K = \alpha^{R_A R_B} \mod \beta$. He then transmits ($\{\alpha^{R_B} \mod \beta\}_P, \{c_B\}_K$) to Alice.

3. Alice calculates $K$ and uses it to decrypt $C_B$, then selects $c_A$ and sends ($\{c_A, c_B\}_K$) to Bob.

4. Bob uses $K$ to decrypt $c_A$ and transmits ($\{c_A\}_K$) to Alice.

  − Why It Works
    * The message encrypted by the password is indistinguishable from a random number.
    * Even if an attacker guesses the correct password, all he gets is a public key — not enough to recover the session key.
    * An attacker must go all the way through a cryptanalytic attack to confirm a putative password.

  − Implementation Subtleties
    * When to encrypt?
      * Depending on the protocol, one of the two encryptions with $P$ can be omitted: Bob isn’t authenticating himself at this point, and Alice can only calculate the session key if she knows $P$. 
This can help to avoid leaking information about $P$, which could be used to recover $K$ without a full cryptanalytic attack.

However, if done incorrectly, this can compromise the protocol.

See [BM92] for details.

* Avoiding easy guess elimination
  * If the scheme used to encrypt the public key requires an odd key, the attacker can eliminate half of the passwords, just by seeing if a password guess encrypts to an even number. (You can send $(n + 1)/2$ in this case).
  * Similarly, if the scheme requires a prime key, you might send it unencrypted to avoid leaking information.

* Examples
  * RSA: see [BM92]
  * ElGamal: see [BM92]

- Attacks on EKE
  * Number Theoretic Attacks [Pat97]
  * Parallel Session Attack [CJ97]

- Other Topics
  - Other Secure Password Protocols (SPPs)
    * SPEKE: Simple Password Exponential Key Exchange (Jablon)
    * SRP: Simple Remote Password (Wu)
    * PDM: Password-Derived Moduli (Kaufman and Perlman)
  - Augmented SPPs
    * Basic ideas
      * Utilize signatures as well as public-key encryption.
      * Protect Alice from man-in-the-middle attacks caused by password file compromise.
    * Augmented Encrypted Key Exchange (AEKE): an extension of the protocol given above:
      1. Alice and Bob share knowledge of $\alpha$ and $\beta$, the Diffie-Hellman base and modulus, respectively, and $V_P$, a key deterministically generated from Alice’s (potentially weak) password, which is used as both the verification key for a signature algorithm (denoted $[M]_{V_P}$) and the encryption key for a symmetric cipher (denoted $\{M\}_{V_P}$).
      2. Alice selects a random $R_A$ and transmits (“Alice”, $\{\alpha^{R_A} \mod \beta\}_P$) to Bob.
      3. Bob selects a random $R_B$ and a challenge $c_B$, and calculates the session key $K = \alpha^{R_A R_B} \mod \beta$. He then transmits ($\{\alpha^{R_B} \mod \beta\}_{V_P}, \{c_B\}_K$) to Alice.
3. Alice calculates $K$ and uses it to decrypt $C_B$, then selects $c_A$ and sends $(\{c_A, c_B\}_K)$ to Bob.
4. Bob uses $K$ to decrypt $c_A$ and transmits $(\{c_A\}_K)$ to Alice.
   * (Up until this point, we’ve done basically the same thing as in EKE; the only change is a requirement that $P$ be usable as $V_P$, a verification key for a signature scheme)
5. Alice sends $(\{[K]_{S_P}\}_K)$, and the protocol concludes successfully only if Bob can verify that $\{\{[K]_{S_P}\}_K\}^{k-1}_{V_P} = K$.
   * For further details and a security analysis, see [BM93]
   * SPEKE and PDM have augmented forms as well, using similar techniques. Choice of whether to use them depends on resources available and security needed.
   * SRP had augmented form built in, which is always used.

- Bibliography


**CJ96** Clark, J. and Jacob, J. ”A survey of authentication protocol literature.” Manuscript. August, 1996.

