**Intiator/Responder anonymity**

**DoS resistance**

**Perfect forward secrecy**

**. . . and other desirable protocol properties**

# Station-to-station protocol

$$
\begin{aligned}
A \longrightarrow B &: g^a \\
B \longrightarrow A &: g^b, Cert_B, \{[\![\{g^b, g^a\}]\!]_{K_B}\}_{g^{ab}} \\
A \longrightarrow B &: Cert_A, \{[\![\{g^a, g^b\}]\!]_{K_A}\}_{g^{ab}}
\end{aligned}
$$

Problem: $A$ learns that $B$ intended to talk to her only after starting to use $g^{ab}$.

# ISO 9798-3 protocol

$$
\begin{aligned}
A \longrightarrow B &: g^a, A, N_A \\
B \longrightarrow A &: [\![ g^a, g^b, N_A, N_B, A ]\!]_{K_B} \\
A \longrightarrow B &: [\![ g^a, g^b, N_A, N_B, B ]\!]_{K_A}
\end{aligned}
$$

- (recall that the signature does not hide the message)
- Adds identities under signature
  - If $A$ has accepted the 2nd message then she knows that $B$ intended to talk to her.
  - If $B$ has accepted the 3rd message then he knows that $A$ intended to talk to him.
- The symmetric key is $H(g^{ab}, A, B, N_A, N_B)$.

# ISO 9798-3 protocol

$$
\begin{aligned}
A \longrightarrow B &: g^a, A, N_A \\
B \longrightarrow A &: [\![ g^a, g^b, N_A, N_B, A ]\!]_{K_B} \\
A \longrightarrow B &: [\![ g^a, g^b, N_A, N_B, B ]\!]_{K_A}
\end{aligned}
$$

- Perfect forward secrecy — if $a$, $b$, $g^{ab}$ are deleted after use, then the leakage of a signing key does not reveal old symmetric keys.
- Vulnerable to DoS — After $B$ receives the first message, he has to
  - store $g^a$, $A$, $N_A$;
  - compute a signature (expensive);
  - (perform a modular exponentiation — compute $g^b$).
    - can be computed ahead-of-time
    - not changed so often
- Not anonymous to a passive eavesdropper.
  - Even if it has no knowledge of network topology.

# Measures against DoS

- To avoid keeping state $\mathbf{S}$
  - Have a long-term symmetric key $K$ known only to yourself.
  - Send $\{\mathbf{S}\}_K$ to the other party.
  - The next message from that party must again contain $\{\mathbf{S}\}_K$.
  - If $\mathbf{S}$ is known to the other party, then encryption can be replaced by a MAC.
- To avoid DoS against computational resources:
  - Perform expensive computations only after the other party must have performed an expensive computation.
  - (the protocol must be designed in such a way)

# Just Fast Keying with initiator privacy

$$
\begin{aligned}
&A \longrightarrow B : g^a, H(N_A)\\
&B \longrightarrow A : H(N_A), N_B, [\![\{g^b\}]\!]_{K_B}, \mathrm{MAC}_{hk_B}(g^b, H(N_A), N_B, IP_A)\\
&A \longrightarrow B : N_A, N_B, \bullet, g^a, g^b, \{K_A, [\![\{g^a, g^b, H(N_A), N_B, K_B\}]\!]_{K_A}\}_{k_{\mathrm{auth}}}\\
&B \longrightarrow A : \{[\![\{g^a, g^b, H(N_A), N_B, K_A\}]\!]_{K_B}\}_{k_{\mathrm{auth}}}
\end{aligned}
$$

$$
\begin{aligned}
k_{\mathrm{auth}} &= H(g^{ab}, H(N_A), N_B, \text{“auth”})\\
k &= H(g^{ab}, H(N_A), N_B, \text{“key”})
\end{aligned}
$$

- $\bullet$ is called a *cookie*.
- Assume that $X$ cannot be legitimately found from $\mathrm{MAC}_K(X)$.

# Design considerations (1)

- Frequency of updating $g^b$ and $[\![\{g^b\}]\!]_{K_B}$ (and $g^a$)
  - A new $g^b$ is computed after a certain time interval, not for each protocol round.
  - Hence $B$ has to keep no state after 2nd message
  - Hence $B$ can respond to the 3rd message multiple times
    - $B$ caches recent pairs of 3rd and 4th messages
    - The cookie is the key for lookup
- Because of cookie, 1st and 3rd messages must come from the same IP-address.
  - If IP was not in the cookie, certain DDoS-attacks were possible.

# Design considerations (2)

- $H(N_A)$ and $N_A$
  - $B$'s first expensive operation is computing $g^{ab}$ after receiving the 3rd message.
  - Before doing it, 3rd message looks like

$$N_A, N_B, \mathrm{MAC}_{hk_B}(g^b, H(N_A), N_B, IP_A), g^a, g^b, \square$$

  - Suppose that $I$ has heard the first two msgs between $A$ and $B$.
  - Suppose that $H(N_A)$ is used instead of $N_A$.
  - $I$ can then construct a message that looks like the one above.

# Password-based authentication

$$A \longrightarrow B : A, pw$$

is very bad.

$$\begin{aligned} B &\longrightarrow A : N_B \\ A &\longrightarrow B : A, N_A, N_B, H(N_A, N_B, pw) \end{aligned}$$

is also bad because of off-line guessing attacks.

# PAK (password-authenticated key exchange)

$$A \longrightarrow B : g^a \cdot H_1(A, B, pw)$$
$$B \longrightarrow A : g^b, H_{2a}(A, B, g^b, \bullet, \left(\frac{\bullet}{H_1(A,B,pw)}\right)^b, pw)$$
$$A \longrightarrow B : H_{2b}(A, B, g^b, \bullet, \bullet, pw)$$

The key is $H_3(A, B, \bullet, \bullet, pw)$

- The blinding/unblinding ability shows the knowledge of the password
- Off-line guessing impossible because of the mask $g^a$
- On-line guessing possible
- Both $A$ and $B$ must store $pw$.

# A protocol with guessing attack

Let $G = \langle g \rangle$ be a group with hard DH problem.
For a user $A$, server $S$ has stored $V_A = h(A, pw_A)$.

$$\begin{aligned} A \longrightarrow S &: A, N_A, g^x \\ S \longrightarrow A &: g^y, N_S \oplus h(g^{xy}, V_A, N_A) \\ A \longrightarrow S &: N_S \end{aligned}$$

$A$ and $S$ have agreed on a common secret $g^{xy}$.
**Exercise.** Attack it.

# An improvement?

Let $G = \langle g \rangle$ be a group with hard DH problem.
For a user $A$, server $S$ has stored $V_A = h(A, pw_A)$.

$$
\begin{aligned}
&A \longrightarrow S : A, N_A, g^x \oplus h(A, V_A) \\
&S \longrightarrow A : g^y \oplus h(A, V_A), N_S \oplus h(g^{xy}, V_A, N_A), g^{xy} \oplus h(g^{xy}, V_A, N_A) \\
&A \longrightarrow S : N_S
\end{aligned}
$$

$A$ and $S$ have agreed on a common secret $g^{xy}$.
**Exercise.** Attack it.

# Encrypted Key Exchange

$A$ and $B$ share a password $pw_{A,B}$.
It is also treated as a symmetric key.

$$
\begin{aligned}
A \longrightarrow B &: \{K^+\}_{pw_{A,B}} \\
B \longrightarrow A &: \{\{[k]\}_{K^+}\}_{pw_{A,B}} \\
A \longrightarrow B &: \{N_A\}_k \\
B \longrightarrow A &: \{N_A, N_B\}_k \\
A \longrightarrow B &: \{N_B\}_k
\end{aligned}
$$

Here $K^+$ is a newly generated public key. $A$ keeps the corresponding private key $K^-$ to herself.

**Exercise.** What if $K^+$ were a symmetric key?

# PAK-X

Server $B$ only has to store $V = g^{pw}$.

$$
\begin{aligned}
A \longrightarrow B &: g^a \cdot H_1(A, B, V) \\
B \longrightarrow A &: g^b, g^c, c \oplus H_{2a}\left(A, B, g^b, \bullet, \left(\frac{\bullet}{H_1(A,B,V)}\right)^b, V^c, V\right) \\
A \longrightarrow B &: H_{2b}(A, B, \langle \text{2nd message} \rangle, \bullet, \bullet, c, V)
\end{aligned}
$$

$A$ has to use $pw$ to recompute $V^c$.