Public Key EXchange

Dan Harkins
Many authenticated key exchange protocols use trusted but uncertified (i.e. “raw”) public keys – IKE, TLS, MQV, etc.

These keys are always established:

“In a manner outside the scope of the protocol”

Problems with gaining trust in “raw” public keys:

– "the main security challenge [to using 'raw' public keys] is how to associate the public key with a specific entity. Without a secure binding between identifier and key, the protocol will be vulnerable to man-in-the- middle attacks.” (RFC 7250)

– Unknown key share attacks are possible if proof-of-possession of the private key is not demonstrated when the public key is exchanged.
Need a standard, programmatic way to exchange "raw" keys that:

- Guarantees the integrity of exchanged keys
- Establishes a secure binding between an identity and the obtained key
- Provides proof-of-possession of corresponding private key
- Is simple, robust, easy to use correctly, and hard to use incorrectly
Use PAKE to obtain secure channel authenticated to an identity
Exchange public keys through secure channel
Provide proof-of-possession of private key

PKEX – Public Key EXchange
– Parlay a simple short-lived word/code/phrase into a trusted and long(er)-lived public key!
– Use case from RFC 8125 (CFRG’s PAKE Requirements)
Goals of PKEX

– Resistant to passive, active, and dictionary attack
– Allows a single public key to be exchanged with a multitude of peers
– Minimal number of primitives
– Upon completion of PKEX each peer trusts the other’s public key:
  ▪ The public key received is the same as the public key the peer sent
  ▪ The peer is in possession of the private analog to the public key
  ▪ The public key is bound to the authenticated identity of the peer
What of PKEX

— Two phases:
  1) “Exchange phase” is SPAKE2
  2) “Commit phase” provides public key, binds it to the PAKE-authenticated exchange, and proves possession of the private key

— Uses role-specific public elements: \( P_i \) – initiator’s; \( P_r \) – responder’s

— Size of prime in group used in PKEX determines primitives:
  • Hash algorithm— \( H() \)
  • Key length of AES-SIV— AE of data \( d \) with key \( k \) and AAD \( s \): \([s]\{d\}_k\)
  • HKDF and HMAC used with \( H() \)

— Element-to-scalar mapping function— \( r = F(R) \)
Given:
- group with generator $G$
- group-specific elements $P_i$ and $P_r$
- Alice shares password pw with “Bob”
- Bob shares password pw with “Alice”

”Exchange Phase”

**Alice**

$x, \ X = x \cdot G$

$Q_i = H(\text{Alice} \mid \text{pw}) \cdot P_i$

$M = X + Q_i$

Alice, $M$ ----->

**Bob**

$y, \ Y = y \cdot G$

$Q_r = H(\text{Bob} \mid \text{pw}) \cdot P_r$

$Q_i = H(\text{Alice} \mid \text{pw}) \cdot P_i$

$X' = M - Q_i$

$N = Y + Q_r$

<--------Bob, $N$

$Q_r = H(\text{Bob} \mid \text{pw}) \cdot P_r$

$Y' = N - Q_r$

$z = \text{HKDF}(F(x \cdot Y'), \text{Alice} \mid \text{Bob} \mid F(M) \mid F(N) \mid \text{pw})$

$z = \text{HKDF}(F(y \cdot X'), \text{Alice} \mid \text{Bob} \mid F(M) \mid F(N) \mid \text{pw})$
Given:
- Alice has identity key a/A
- Bob has identity key b/B

"Commit Phase"

Alice

\[
\begin{align*}
  u &= \text{HMAC}(F(a*Y'), \text{Alice} | F(A) | F(Y') | F(X)) \\
  [0]\{ A, u \}_z &\longrightarrow
\end{align*}
\]

Bob

If (SIV-decrypt returns fail) fail
If (A not valid element) fail
\[
\begin{align*}
  u' &= \text{HMAC}(F(y*A), \text{Alice} | F(A) | F(Y) | F(X')) \\
  \text{If (u' != u) fail}
\end{align*}
\]

\[
\begin{align*}
  v &= \text{HMAC}(F(b*X'), \text{Bob} | F(B) | F(X') | F(Y)) \\
  \text{If (SIV-decrypt returns fail) fail}
\end{align*}
\]

If (B not valid element) fail
\[
\begin{align*}
  v' &= \text{HMAC}(F(x*B), \text{Bob} | F(B) | F(X) | F(Y')) \\
  \text{If (v' != v) fail}
\end{align*}
\]

\[
\begin{align*}
  [1]\{ B, v \}_z &\longrightarrow
\end{align*}
\]
Upon successful completion of PKEX...

✓ Alice possesses the public key Bob sent
✓ Alice has assurance that this is really Bob’s key
✓ Alice knows Bob possesses his private key

(Ditto for Bob to Alice)

Raw Public Keys are now trusted for use!

Privacy note: While Alice and Bob expose their identities during PKEX, their public “identity keys”, to which their identities are bound, are not exposed which could afford a modicum of privacy to their subsequent use in some other AKM protocol.
What Now?

draft-harkins-pkex-04 is latest-and-greatest
  – Three independent interoperable implementations
  – Received some cryptanalysis
  – Appendix contains role-specific elements for 6 popular ECC groups and 4 popular FFC groups

I’d like the -04 draft to be adopted by CFRG as a work item