Derandomized PACE with Mutual Authentication

Adam Bobowski and Mirosław Kutyłowski

Wrocław University of Science and Technology, Poland

NSS 2019, Sapporo

ICAO

PACE protocol is obligatory for newer biometric passports.

For older BAC is used but the algorithm is obsolete.

> 490 million

ePassports in circulation [1]

EU

Starting from 2021 member countries of EU will not be allowed to issue official Identity Documents without PACE.

> 510 million

the population of the European Union as of 2019 [2]

[1] from www.icao.int
[2] from www.ec.europa.eu

Password Authenticated Key Exchange
On example of ePassport:

1. The passport stores the password, the reader gets the password by optically scanning the CAN code number printed in Machine Readable Zone.
2. The passport and the reader run a key exchange protocol but at the same time make sure that they use the same password.
3. It is infeasible for an eavesdropper to deduce the password, even for an active adversary.
4. A malicious reader having no access to the password can start the protocol, but it will fail leaving no usable information, unless it is using the right password.

How PACE works (generally speaking)

Alice (eID chip)
Holds password

Bob (reader)
Reads password from input

Alice: Chooses s at random
Sends s encrypted with password derived key.

Bob: Decrypts s with key derived from password

1st Diffie-Hellman key exchange (in g)
(Alice and Bob choose x_A and x_B at random)

Derivation of new base point ($\tilde{g} = g^{xaxb}g^{s}$)

2nd Diffie-Hellman key exchange (in $\tilde{g}$)
(Alice and Bob choose y_A and y_B at random)

Derivation of Enc and MAC keys from 2nd DH.

Verification of computed values.

Randomness is the key problem

- Security assurances of the protocol strongly depends on the quality of random number generator.

- PRNG module might be the weakest link in the physical devices.

  - Entropy source on low-end devices might not be trustworthy.

  - Obtaining good randomness is expensive - (commercial tradeoffs).

What did we do?

- Removed randomness from PACE while maintaining the level of security.

- Introduced option for stronger authentication.
  (pure PACE does not provide strong authentication of the communication parties: Chip Authentication (CAM) and Terminal Authentication have to be executed separately)

- Maintained execution compatibility with original PACE.

How did we do that?

- Replaced each sampling of random values with deterministic operation on established seed ω.

- Added private / public keys for devices.

- Added initialization phase that derives the seed ω based on:
    - the context,
    - password,
    - and verification option (anonymous / non-anonymous).

- Added authentication phase that verifies if correct seed was used (both for anonymous and non-anonymous option).

device A (eID chip)		device B (reader)
holds the keys:		holds the keys:
sk_A (secret), $pk_A = g^{sk_A}$ (public)		sk_B (secret), $pk_B = g^{sk_B}$ (public)
password π		password π (input)
	initialization	
adjust the password π to the context		adjust the password π to the context
determine the seed ω_A		determine the seed ω_B
	PACE initial phase	
$K_\pi := H(\pi \| 0)$		$K_\pi := H(\pi \| 0)$
~~choose s at random~~		
$s := H_q(\omega_A \| 4)$		
$z := \mathrm{Enc}(K_\pi, s)$	$\xrightarrow{\mathcal{G}, z}$	abort if $\mathcal{G}$ incorrect
		$s := \mathrm{Dec}(K_\pi, z)$
	DH2Point Start	
~~choose x_A at random~~		~~choose x_B at random~~
$x_A := H_q(\omega_A \| 5)$		$x_B := H_q(\omega_B \| 6)$
$X_A := g^{x_A}$	$\xleftarrow{X_B}$	$X_B := g^{x_B}$
	$\xrightarrow{X_A}$	
$h := X_B^{x_A}$		$h := X_A^{x_B}$
abort if $h = 1$		abort if $h = 1$
$\hat{g} := h \cdot g^s$		$\hat{g} := h \cdot g^s$
	DH2Point End	

Party A		Party B
~~choose y_A at random~~		~~choose y_B at random~~
$y_A := H_q(\omega_A \| 7)$		$y_B := H_q(\omega_B \| 8)$
$Y_A := \hat{g}^{y_A}$	$\xleftarrow{Y_B}$	$Y_B := \hat{g}^{y_B}$
	$\xrightarrow{Y_A}$	
$K := {Y_B}^{y_A}$		$K := {Y_A}^{y_B}$
$K_{\text{Enc}} := H(K \| 1)$		$K_{\text{Enc}} := H(K \| 1)$
$K_{\text{MAC}} := H(K \| 2)$		$K_{\text{MAC}} := H(K \| 2)$
$K'_{\text{MAC}} := H(K \| 3)$		$K'_{\text{MAC}} := H(K \| 3)$
$K'_{\text{Enc}} := H(K \| 4)$		$K'_{\text{Enc}} := H(K \| 4)$
$T_A := \text{MAC}(K'_{\text{MAC}}, (Y_B, \mathcal{G}, \hat{g}))$		$T_B := \text{MAC}(K'_{\text{MAC}}, (Y_A, \mathcal{G}, \hat{g}))$
	$\xleftarrow{T_B}$	
abort if T_B incorrect	$\xrightarrow{T_A}$	abort if T_A incorrect

Authentication procedure

authenticating the parties based on the the public keys pk_A, pk_B and the protocol transcript

Two Modes of Authentication

Anonymous Authentication
Devices do not exchange their public keys.

Initialization:
basepoint g is derived from context and seed is set as $\omega = g^{sk}$

Authentication:
Exchange ω and proof the knowledge of exponent

Non-Anonymous Authentication
Devices do exchange their public keys.

Initialization:
seed ω is derived from context and DH on public keys of other party
(should be the same for both parties)

Authentication:
Implicitly (check if values received were computed correctly)

Conclusion

Goal:

Remove necessity of random number generator and thereby reduce the cost of the chip, while maintaining the level of security.

Result:

Derandomized and PACE compatible protocols that are interoperable with original.

Additional better verifiability properties, as well as chip and terminal authentication.

Thanks for Your attention!

Contact:

adam.bobowski@pwr.edu.pl

Wrocław University of Science and Technology