Protection of Digital Images from Personal Identity Documents

Przemysław Kubiak Mirosław Kutyłowski
Wojciech Wodo

CECC 2013, June 26-28, Telč

Motivation

- Electronic layer of e-ID may store a high resolution face image of the document holder – more detailed than the image printed on the document.

Motivation

- Electronic layer of e-ID may store a high resolution face image of the document holder – more detailed than the image printed on the document.
- The strategy applied in particular by biometric passports is to present not only raw data, but also a signature of the document issuer for those data.

Motivation

- Electronic layer of e-ID may store a high resolution face image of the document holder – more detailed than the image printed on the document.
- The strategy applied in particular by biometric passports is to present not only raw data, but also a signature of the document issuer for those data.
  In this way during an inspection we may become convinced that the image presented originates from the document issuer and has not been replaced even if chip security of e-ID has been broken.

Motivation

- Electronic layer of e-ID may store a high resolution face image of the document holder – more detailed than the image printed on the document.
- The strategy applied in particular by biometric passports is to present not only raw data, but also a signature of the document issuer for those data.
  In this way during an inspection we may become convinced that the image presented originates from the document issuer and has not been replaced even if chip security of e-ID has been broken.
- Once the signed data is shown to a second party, the owner of e-ID has no further control over who has access to it. In particular, this data can be sold to third parties.

System goals

- Once a face image is presented by an e-ID, then a customized signature of the document issuer is attached.

System goals

- Once a face image is presented by an e-ID, then a customized signature of the document issuer is attached.
- The signature indicates the recipient of the signature.

System goals

- Once a face image is presented by an e-ID, then a customized signature of the document issuer is attached.
- The signature indicates the recipient of the signature.
- Issuer's signing key is not copied on the e-ID!

System goals

- Once a face image is presented by an e-ID, then a customized signature of the document issuer is attached.
- The signature indicates the recipient of the signature.
- Issuer's signing key is not copied on the e-ID!
- The authority issuing the e-ID documents cannot create clone documents and customized signatures in order to accuse a certain party for violations of personal data protection.

Assumptions about e-ID chips

- We assume that keys generated privately on the chip can be read by the e-ID issuer as long as the key generation process takes place in environment controlled by the issuer.

Assumptions about e-ID chips

- We assume that keys generated privately on the chip can be read by the e-ID issuer as long as the key generation process takes place in environment controlled by the issuer.
- However, keys generated on the chip when the e-ID is in control of the owner are neither predictable for the e-ID issuer nor they leak from the e-ID.

Assumptions about e-ID chips

- We assume that keys generated privately on the chip can be read by the e-ID issuer as long as the key generation process takes place in environment controlled by the issuer.
- However, keys generated on the chip when the e-ID is in control of the owner are neither predictable for the e-ID issuer nor they leak from the e-ID.

The assumptions above reflects the setting where:

- the chip vendor does not collude with the document issuer,

Assumptions about e-ID chips

- We assume that keys generated privately on the chip can be read by the e-ID issuer as long as the key generation process takes place in environment controlled by the issuer.
- However, keys generated on the chip when the e-ID is in control of the owner are neither predictable for the e-ID issuer nor they leak from the e-ID.

The assumptions above reflects the setting where:

- the chip vendor does not collude with the document issuer,
- the issuer has access to technologies that with phisical access to the chip may break security means on the chip and can access all relevant data on it.

Solutions

- Two protocols are presented in the paper:
  - a symmmetric one,
  - an asymmetric one.

Solutions

- Two protocols are presented in the paper:
  - a symmmetric one,
  - an asymmetric one.
- The first one is well suited to weak devices (like e-passports with BAC), but resolution of indication of signature recipient is limited (to say 2^{10} classes of recipients).
- The second one requires more capable chips (like e-passports with EAC), but the resolution mentioned above is unlimited.

The Asymmetric Solution
The Main Mechanism

- As usually, the datagroups D and the signature of the document issuer are presented to the verifier (*for simplicity we assume that all datagroups are revealed to a verifier*).

The Asymmetric Solution
The Main Mechanism

- As usually, the datagroups D and the signature of the document issuer are presented to the verifier (*for simplicity we assume that all datagroups are revealed to a verifier*).
- But the chip of e-ID attaches a tag to the pair: (the data groups, the signature).

The Asymmetric Solution
The Main Mechanism

- As usually, the datagroups D and the signature of the document issuer are presented to the verifier (*for simplicity we assume that all datagroups are revealed to a verifier*).
- But the chip of e-ID attaches a tag to the pair: (the data groups, the signature).
- The point is that without the tag signature verification is infeasible, and that the tag indicates the intended verifier.

The Asymmetric Solution

The Main Building Block - Schnorr-like Proof of EDL

Let $\langle g \rangle$ be of prime order q. Let DDHP be hard in $\langle g \rangle$.

The prover performs the following steps:

1. generate r at random,
2. $k := g^r$, $\ell := h^r$,
3. $e := H(k, \ell, g, h, a, b, m)$, where m is some message
4. $s := r + ex \bmod q$,
5. send (e, s) to the verifier.

The verifier performs the following steps:

1. $k' := g^s / a^e$,
2. $\ell' := h^s / b^e$,
3. $e' := H(k', \ell', g, h, a, b, m)$,
4. return $\mathtt{ok}$ if $e = e'$.

The Protocol

- The system is supported by a card management system called below CAMS.
- We refer also to standard protocols for chip authentication (Chip Authentication or ChA) and authenticating terminals (Terminal Authentication or TA) [BSI TR-03110].

The Protocol
Document personalization

For each single identity document the following steps are executed by issuing authority:

The Protocol
Document personalization

For each single identity document the following steps are executed by issuing authority:

1. All but two data groups for the e-ID are completed in advance, and are stored in some registry on the side of CAMS.

The Protocol
Document personalization

For each single identity document the following steps are executed by issuing authority:

1. All but two data groups for the e-ID are completed in advance, and are stored in some registry on the side of CAMS.
2. The data groups are copied to the chip of e-ID.

The Protocol
Document personalization

For each single identity document the following steps are executed by issuing authority:

1. All but two data groups for the e-ID are completed in advance, and are stored in some registry on the side of CAMS.
2. The data groups are copied to the chip of e-ID.
3. The private key and the corresponding public key for ChA are generated by the e-ID chip.

The Protocol
Document personalization

For each single identity document the following steps are executed by issuing authority:

1. All but two data groups for the e-ID are completed in advance, and are stored in some registry on the side of CAMS.
2. The data groups are copied to the chip of e-ID.
3. The private key and the corresponding public key for ChA are generated by the e-ID chip.
4. The ChA public key is copied to the data groups (i.e., to a copy stored locally on the e-ID chip as well to a copy stored in the registry of CAMS).

The Protocol
Document personalization

For each single identity document the following steps are executed by issuing authority:

1. All but two data groups for the e-ID are completed in advance, and are stored in some registry on the side of CAMS.
2. The data groups are copied to the chip of e-ID.
3. The private key and the corresponding public key for ChA are generated by the e-ID chip.
4. The ChA public key is copied to the data groups (i.e., to a copy stored locally on the e-ID chip as well to a copy stored in the registry of CAMS).
5. The e-ID chip enters a “red” state

The Protocol
Document personalization - results

- The data groups are still not authenticated by the issuing authority.

The Protocol
Document personalization - results

- The data groups are still not authenticated by the issuing authority.
- The e-ID is in a “red” state, which means that all functions of the chip are blocked – only Terminal Authentication and Chip Authentication with terminals of CAMS are allowed.

The Protocol

Document personalization - results

- The data groups are still not authenticated by the issuing authority.
- The e-ID is in a “red” state, which means that all functions of the chip are blocked – only Terminal Authentication and Chip Authentication with terminals of CAMS are allowed.
- When the e-ID is in hands of its owner, it must be unblocked.

The Protocol

Unblocking the chip - Phase I

In a private environment the owner connects to a service of CAMS and after execution of TA+ChA:

The Protocol

Unblocking the chip - Phase I

In a private environment the owner connects to a service of CAMS and after execution of TA+ChA:

1. The e-ID chip generates its private key $\tilde{x}$ for tagging, and computes $\tilde{a} = g^{\tilde{x}}$, where g is fixed in the system (the same for all users).

The Protocol

Unblocking the chip - Phase I

In a private environment the owner connects to a service of CAMS and after execution of TA+ChA:

1. The e-ID chip generates its private key $\tilde{x}$ for tagging, and computes $\tilde{a} = g^{\tilde{x}}$, where g is fixed in the system (the same for all users).
2. Key $\tilde{a}$ is written in the remaining empty data group, both in the e-ID chip and in the CAMS registry.

The Protocol

Unblocking the chip - Phase I

In a private environment the owner connects to a service of CAMS and after execution of TA+ChA:

1. The e-ID chip generates its private key $\tilde{x}$ for tagging, and computes $\tilde{a} = g^{\tilde{x}}$, where g is fixed in the system (the same for all users).
2. Key $\tilde{a}$ is written in the remaining empty data group, both in the e-ID chip and in the CAMS registry.
3. The e-ID chip and CAMS each compute $\tilde{h} = H_g(D)$, where H_g is a hash function with the image included in the group generated by g.

The Protocol

Unblocking the chip - Phase I

In a private environment the owner connects to a service of CAMS and after execution of TA+ChA:

1. The e-ID chip generates its private key $\tilde{x}$ for tagging, and computes $\tilde{a} = g^{\tilde{x}}$, where g is fixed in the system (the same for all users).
2. Key $\tilde{a}$ is written in the remaining empty data group, both in the e-ID chip and in the CAMS registry.
3. The e-ID chip and CAMS each compute $\tilde{h} = H_g(D)$, where H_g is a hash function with the image included in the group generated by g.
4. The e-ID chip computes $\tilde{b} = \tilde{h}^{\tilde{x}}$ and sends $\tilde{b}$ to CAMS.

The Protocol

Unblocking the chip - Phase I

In a private environment the owner connects to a service of CAMS and after execution of TA+ChA:

1. The e-ID chip generates its private key $\tilde{x}$ for tagging, and computes $\tilde{a} = g^{\tilde{x}}$, where g is fixed in the system (the same for all users).
2. Key $\tilde{a}$ is written in the remaining empty data group, both in the e-ID chip and in the CAMS registry.
3. The e-ID chip and CAMS each compute $\tilde{h} = H_g(D)$, where H_g is a hash function with the image included in the group generated by g.
4. The e-ID chip computes $\tilde{b} = \tilde{h}^{\tilde{x}}$ and sends $\tilde{b}$ to CAMS.
5. The e-ID chip and CAMS execute the Schnorr-like ZKP for equality of discrete logarithms for $\tilde{a}, \tilde{b}$ and the corresponding bases $g, \tilde{h}$ (m is chosen to be the string "CAMS").

The Protocol

Unblocking the chip - Phase I

In a private environment the owner connects to a service of CAMS and after execution of TA+ChA:

1. The e-ID chip generates its private key $\tilde{x}$ for tagging, and computes $\tilde{a} = g^{\tilde{x}}$, where g is fixed in the system (the same for all users).
2. Key $\tilde{a}$ is written in the remaining empty data group, both in the e-ID chip and in the CAMS registry.
3. The e-ID chip and CAMS each compute $\tilde{h} = H_g(D)$, where H_g is a hash function with the image included in the group generated by g.
4. The e-ID chip computes $\tilde{b} = \tilde{h}^{\tilde{x}}$ and sends $\tilde{b}$ to CAMS.
5. The e-ID chip and CAMS execute the Schnorr-like ZKP for equality of discrete logarithms for $\tilde{a}, \tilde{b}$ and the corresponding bases $g, \tilde{h}$ (m is chosen to be the string "CAMS").
6. The e-ID chip enters a "yellow" state.

The Protocol
Phase I – Results

We have the following mappings:

$$g \longrightarrow \tilde{a} = g^{\tilde{x}}$$

$$\tilde{h} \longrightarrow \tilde{b} = \tilde{h}^{\tilde{x}}$$

where

- g is fixed for all users
- $\tilde{h}$ is calculated from the data groups: $\tilde{h} = H_g(D)$
- $\tilde{a}$ is written in D

The Protocol
Unblocking the chip - Phase II

On the side of CAMS:

The Protocol
Unblocking the chip - Phase II

On the side of CAMS:

1. User's data groups from CAMS's registry are transferred, together with the ZKP of EDL, to the document issuing authority.

The Protocol

Unblocking the chip - Phase II

On the side of CAMS:

1. User's data groups from CAMS's registry are transferred, together with the ZKP of EDL, to the document issuing authority.
2. The document issuing authority verifiers the proof and generates a signature $Sign(\tilde{b})$ under $\tilde{b}$.

The Protocol

Unblocking the chip - Phase II

On the side of CAMS:

1. User's data groups from CAMS's registry are transferred, together with the ZKP of EDL, to the document issuing authority.
2. The document issuing authority verifiers the proof and generates a signature $Sign(\tilde{b})$ under $\tilde{b}$.
3. $Sign(\tilde{b})$ is transferred back to CAMS's registry.

The Protocol

Unblocking the chip - Phase II

On the side of CAMS:

1. User's data groups from CAMS's registry are transferred, together with the ZKP of EDL, to the document issuing authority.
2. The document issuing authority verifiers the proof and generates a signature $Sign(\tilde{b})$ under $\tilde{b}$.
3. $Sign(\tilde{b})$ is transferred back to CAMS's registry.

On the side of e-ID:

The Protocol

Unblocking the chip - Phase II

On the side of CAMS:

1. User's data groups from CAMS's registry are transferred, together with the ZKP of EDL, to the document issuing authority.
2. The document issuing authority verifiers the proof and generates a signature $Sign(\tilde{b})$ under $\tilde{b}$.
3. $Sign(\tilde{b})$ is transferred back to CAMS's registry.

On the side of e-ID:

1. If an e-ID is in the "yellow" state, then any time the e-ID is used it tells the middle-ware to connect to CAMS's service to fetch $Sign(\tilde{b})$.

The Protocol

Unblocking the chip - Phase II

On the side of CAMS:

1. User's data groups from CAMS's registry are transferred, together with the ZKP of EDL, to the document issuing authority.
2. The document issuing authority verifiers the proof and generates a signature $Sign(\tilde{b})$ under $\tilde{b}$.
3. $Sign(\tilde{b})$ is transferred back to CAMS's registry.

On the side of e-ID:

1. If an e-ID is in the "yellow" state, then any time the e-ID is used it tells the middle-ware to connect to CAMS's service to fetch $Sign(\tilde{b})$.
2. Once $Sign(\tilde{b})$ is fetched, the e-ID switches from the "yellow" state to the "green" one ("regular usage").

The Protocol
Phase II – Results

We have the following mappings:

$$g \longrightarrow \tilde{a} = g^{\tilde{x}}$$

$$\tilde{h} \longrightarrow \tilde{b} = \tilde{h}^{\tilde{x}}$$

where

- g is fixed for all users
- $\tilde{h}$ is calculated from the data groups: $\tilde{h} = H_g(D)$
- $\tilde{a}$ is written in D
- The e-ID has $Sign(\tilde{b})$ under $\tilde{b}$.

The Protocol
Data Group Authentication

To execute this part the e-ID must be in “green” state. After completion of TA+ChA:

The Protocol
Data Group Authentication

To execute this part the e-ID must be in “green” state. After completion of TA+ChA:

1. The e-ID chip sends D and $Sign(\tilde{b})$ to the terminal.

The Protocol
Data Group Authentication

To execute this part the e-ID must be in “green” state. After completion of TA+ChA:

1. The e-ID chip sends D and $Sign(\tilde{b})$ to the terminal.
2. The terminal reads $\tilde{a}$ from D and computes $\tilde{h} = H_g(D)$.

The Protocol
Data Group Authentication

To execute this part the e-ID must be in “green” state. After completion of TA+ChA:

1. The e-ID chip sends D and $Sign(\tilde{b})$ to the terminal.
2. The terminal reads $\tilde{a}$ from D and computes $\tilde{h} = H_g(D)$.
3. The e-ID chip computes $\tilde{h} = H_g(D)$ and $\tilde{b} = \tilde{h}^{\tilde{x}}$ and sends $\tilde{b}$ to the terminal (now both sides know the tuple $(\tilde{a}, \tilde{b}, g, \tilde{h})$ and $Sign(\tilde{b})$, but the link between $\tilde{h}$ and $\tilde{b}$ must be proven by the e-ID chip).

The Protocol
Data Group Authentication

To execute this part the e-ID must be in “green” state. After completion of TA+ChA:

1. The e-ID chip sends D and $Sign(\tilde{b})$ to the terminal.
2. The terminal reads $\tilde{a}$ from D and computes $\tilde{h} = H_g(D)$.
3. The e-ID chip computes $\tilde{h} = H_g(D)$ and $\tilde{b} = \tilde{h}^{\tilde{x}}$ and sends $\tilde{b}$ to the terminal (now both sides know the tuple $(\tilde{a}, \tilde{b}, g, \tilde{h})$ and $Sign(\tilde{b})$, but the link between $\tilde{h}$ and $\tilde{b}$ must be proven by the e-ID chip).
4. Both parties execute ZKP for EDL for $\tilde{a}, \tilde{b}$ and the corresponding bases $g, \tilde{h}$. Schnorr-like protocol is used for m being a string identifying the verifier.

The Protocol
Security

- The exponentiation $\tilde{h}^{\tilde{x}}$, where $\tilde{h} = H_g(D)$, resembles BLS signature scheme. However, if $\langle g \rangle$ would be a pairing friendly group, no ZKP-EDL would be necessary, because equality could immediately be checked with pairing (but we have assumed that DDHP is hard in $\langle g \rangle$).
  Thus augmenting the exponentiation with ZKP-EDL we use a kind of an analog of BLS signature scheme in pairing unfriendly groups.

The Protocol
Security

- The exponentiation $\tilde{h}^{\tilde{x}}$, where $\tilde{h} = H_g(D)$, resembles BLS signature scheme. However, if $\langle g \rangle$ would be a pairing friendly group, no ZKP-EDL would be necessary, because equality could immediately be checked with pairing (but we have assumed that DDHP is hard in $\langle g \rangle$).
  Thus augmenting the exponentiation with ZKP-EDL we use a kind of an analog of BLS signature scheme in pairing unfriendly groups.
- Since D is of the form $(g^{\tilde{x}}, M)$, where M are some data, we obtain a kind of a self-signed certificate of the public key $\tilde{a} = g^{\tilde{x}}$.
  The document issuing authority makes signature $Sign(\tilde{b})$ under the BLS-like "signature" value $\tilde{b} = \tilde{h}^{\tilde{x}}$.

The Protocol
Security

- The exponentiation $\tilde{h}^{\tilde{x}}$, where $\tilde{h} = H_g(D)$, resembles BLS signature scheme. However, if $\langle g \rangle$ would be a pairing friendly group, no ZKP-EDL would be necessary, because equality could immediately be checked with pairing (but we have assumed that DDHP is hard in $\langle g \rangle$).
  Thus augmenting the exponentiation with ZKP-EDL we use a kind of an analog of BLS signature scheme in pairing unfriendly groups.
- Since D is of the form $(g^{\tilde{x}}, M)$, where M are some data, we obtain a kind of a self-signed certificate of the public key $\tilde{a} = g^{\tilde{x}}$.
  The document issuing authority makes signature $Sign(\tilde{b})$ under the BLS-like "signature" value $\tilde{b} = \tilde{h}^{\tilde{x}}$.
- **Problem:** *is it feasible to change M and tune $\tilde{x}$ accordingly in such a way that $\tilde{b}$ remains unchanged?*

The Protocol
Security

- When we try to change M to M' we search for $x' \in \mathbb{Z}_q^*$ yielding a collision:

$$\tilde{b}^{(x')^{-1}} = H_g(g^{x'}, M').$$

The Protocol
Security

- When we try to change M to M' we search for $x' \in \mathbb{Z}_q^*$ yielding a collision:

$$\tilde{b}^{(x')^{-1}} = H_g(g^{x'}, M').$$

- Probability of such an event is not greater than probability of the following collision

$$\tilde{b}^{(x')^{-1}} = H_g(y, M'),$$

where x', y could be independently chosen.

The Protocol
Security

- When we try to change M to M' we search for $x' \in \mathbb{Z}_q^*$ yielding a collision:

$$\tilde{b}^{(x')^{-1}} = H_g(g^{x'}, M').$$

- Probability of such an event is not greater than probability of the following collision

$$\tilde{b}^{(x')^{-1}} = H_g(y, M'),$$

where x', y could be independently chosen.

- But the latter collision occurs no more frequently than the collision

$$\tilde{b}^{(x')^{-1}} = H_g(\tilde{M}),$$

where $\tilde{M}$ could be any bitstring.

The Protocol
Security

In the random oracle model for H_g probability of the last event results from the birthday paradox in two rooms setting. Let fix parameter $\gamma \in (0, 1)$:

The Protocol
Security

In the random oracle model for H_g probability of the last event results from the birthday paradox in two rooms setting. Let fix parameter $\gamma \in (0, 1)$:

- Provided that in each single choice of $(x', \tilde{M})$ an element $\tilde{b}^{(x')^{-1}} \in \mathrm{Im}(H_g)$, the number of choices $(x', \tilde{M})$ yelding the collision with probability no smaller than γ is equal to $c_\gamma \cdot \sqrt{|\mathrm{Im}(H_g)|}$, where constant c_γ is dependent of γ.

The Protocol
Security

In the random oracle model for H_g probability of the last event results from the birthday paradox in two rooms setting. Let fix parameter $\gamma \in (0,1)$:

- Provided that in each single choice of $(x', \tilde{M})$ an element $\tilde{b}^{(x')^{-1}} \in \mathrm{Im}(H_g)$, the number of choices $(x', \tilde{M})$ yelding the collision with probability no smaller than γ is equal to $c_\gamma \cdot \sqrt{|\mathrm{Im}(H_g)|}$, where constant c_γ is dependent of γ.
- Since $x', \tilde{M}$ could be chosen independently, the expected number of choices of $(x', \tilde{M})$ to obtain the collision with probability no smaller than γ, equals in the random oracle model for H_g to

$$\frac{c_\gamma \cdot \sqrt{|\mathrm{Im}(H_g)|}}{\Pr\left(\tilde{b}^{(x')^{-1}} \in \mathrm{Im}(H_g)\right)}.$$

Thanks for your attention!

This research was initiated under support of Foundation for Polish Science.