Seminar Biometry & Security, Winter 2009/2010

An Authentication Protocol with crypted Biometric Data

Based on Bringer et. Al.

Andreas Dresen
31.01.2010

Inhaltsverzeichnis

1. Introduction

Authentication protocols can work in different ways. One possibility is through symmetric or asymmetric key pairs. Another way is to use a password. But they can be broken, if the password is not long enough or is spied out by a third person. Also protocols using key pairs can be broken, if the used group is to weak or the keys itself.
A possible solution is authentication protocols that use biometric templates of the user like iris scans or fingerprints. Normally it is impossible respectively nearly impossible to forge such templates. Also you cannot forget your password because you have your template always with you. To guarantee that the protocol works fine, it must ensure some assumptions; see subparagraph 1.1.
This new authentication protocol described here is based on a former Protocol by (Bringer, et al., 2007). In this old protocol they use a normal Private Information Retrieval (PIR) scheme for achieving the privacy of biometric data of users. The new protocol use, instead of the old one, Lipmaa's PIR scheme. It also adds Secure Sketches and the homomorphic properties of the Goldwasser-Micali cryptosystem.

1.1 A normal authentication protocol

A normal authentication protocol consists of two phases. First the enrollment phase where the user measures and stores his biometric template for the first time. Second the verification phase where the user authenticates him at the database to get access to a secured place or an encrypted document for example.

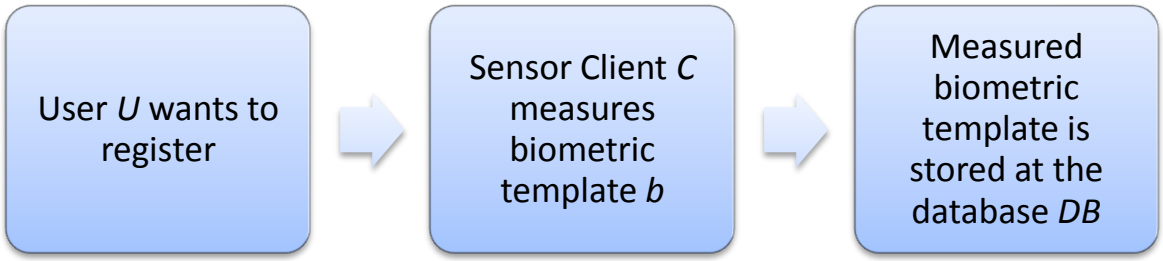

Image 1, Enrollment phase

In the authentication phase, matching algorithms check, if the new biometric template is similar to the stored one or not. This is done by threshold. Is the difference of both templates below the threshold the authentication works fine else the request is rejected.

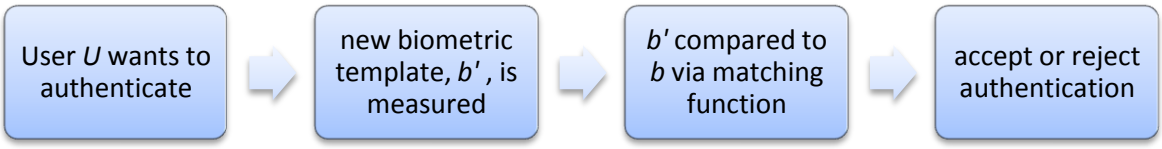

Image 2, Verification phase

Further there has to be several assumptions to secure that the scheme works in the way it should work. First is the classical liveness assumption:

Assumption 1: *We assume that, with a high probability, the biometric template captured by the sensor and used in the system is from a living human user. In other words, it is difficult to produce a fake biometric template that can be accepted by the sensor.*

Assumption 2: *With respect to the authentication service, service provider is trusted by human users to make the right decision, and database is trusted by human users and the service provider to store and provide the right biometric information. Only an outside adversary may try to impersonate a hones human user.*

Assumption 3: *With respect to privacy concerns, both service provider and database are assumed to be malicious which means they may deviate from the protocol specification, but they will not collude. In reality, an outside adversary may also pose threats to the privacy concerns, however, it has no more advantage than a malicious system component.*

(Bringer, et al., 2008)

2. Sketches

2.1 Secure Sketches

Secure Sketches allow a strict disconnection between the biometric data of the users and on the other hand short-term data, generated only for the authentication of a user at the service provider. A secure sketches scheme allows also restoring a hidden value from any element close to this hidden value. To do this you imagine the differences between two captured biometric templates as errors over a codeword. This is possible with the hamming distance for example. With the hamming distance you are able to recover a string $w \in \mathcal{H}$ from a string $w' \in \mathcal{H}$, which is close to w. This is feasible through a known data P that does not reveal a lot of information about w. With all these it is already achievable to construct an authentication protocol.

In the registration phase we store $P = SS_C(w) = c \oplus w$ and the hash value $H(c)$, where *c* is random codeword and *H* is a crpytographic hash function. In case of authenticate someone first try to correct the likely corrupted codeword $w' \oplus P = c \oplus (w' \oplus w)$ and if a codeword c' is received, check the hash value $H(c') = H(c)$. To keep off an attacker from doing an entire search of codewords and so restore biometric data, the size of the code shall not be to small. But with a big dimension for the code too it is better to add additional tools to increase the security.

2.2 Goldwasser-Micali Scheme

This is only a short review of the Goldwasser-Micali scheme containing the different parts of the algorithm. First the algorithm *Gen* for the key generation.

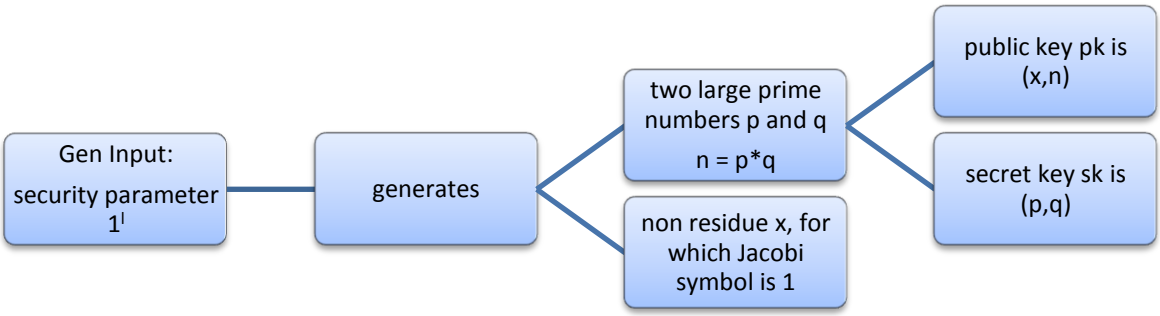

Image 3, Key generation of Goldwasser-Micali scheme

The encryption algorithm Enc takes a message m that consists of a single bit and the public key as input. The output is a ciphertext c which is computed in the following way $c = y^2 x^m \ mod\ n$. Variable y is chosen randomly from $\mathbb{Z}_n^*$. For the decryption algorithm Dec the input is the ciphertext c and the private key sk. The output of Dec is the message m, whereby m = 0 if the ciphertext c is a quadratic residue else m = 1. Reason for output always a single bit is that the encryption algorithm can only encrypt one bit at a time. If you have a binary string of length 7 for example, you must run the encryption algorithm 7 times. And also run 7 times the decryption algorithm for decrypting the message.
This scheme is semantically secure if the Quadratic Residue problem is unsolvable. To put it another way an adversary $\mathcal{A}$ has only a negligible advantage in the following game.

$$
\begin{aligned}
&Exp_{\varepsilon,\mathcal{A}}^{IND-CPA} \\
&(sk, pk) \leftarrow Gen(1^l) \\
&(m_0 m_1) \leftarrow \mathcal{A}(pk) \\
&\quad c \quad \leftarrow Enc(m_\beta, pk),\ \beta \overset{R}{\leftarrow} \{0,1\} \\
&\quad \beta' \quad \leftarrow \mathcal{A}(m_0, m_1, c, pk)
\end{aligned}
$$

(Bringer, et al., 2008)

So the advantage of the attacker is at the end of this game the following

$$Adv_{\varepsilon,\mathcal{A}}^{IND-CPA} = \left|Pr\left[Exp_{\varepsilon,\mathcal{A}}^{IND-CPA} = 1|\beta = 1\right] - Pr\left[Exp_{\varepsilon,\mathcal{A}}^{IND-CPA} = 1|\beta = 0\right]\right|$$

(Bringer, et al., 2008)

Furthermore the encryption protocol has homomorphic property for any m, m' that is made of 0's or 1's. The equation is as follows.

$$Dec(Enc(m, pk) \times Enc(m', pk), sk) = m \oplus m'$$

This property is later used in the authentication protocol via the encrypted sketches.

2.3 Encrypted Sketches

In case of normal sketches everybody can check, if a biometric template is contained in the database or not, assumed he has access to the database. To prevent this and because of privacy concerns, store the biometric data encrypted in the database and also do all computations with the encrypted data. This can be reached through the use of the Goldwasser-Micali scheme, described in the previous subparagraph.

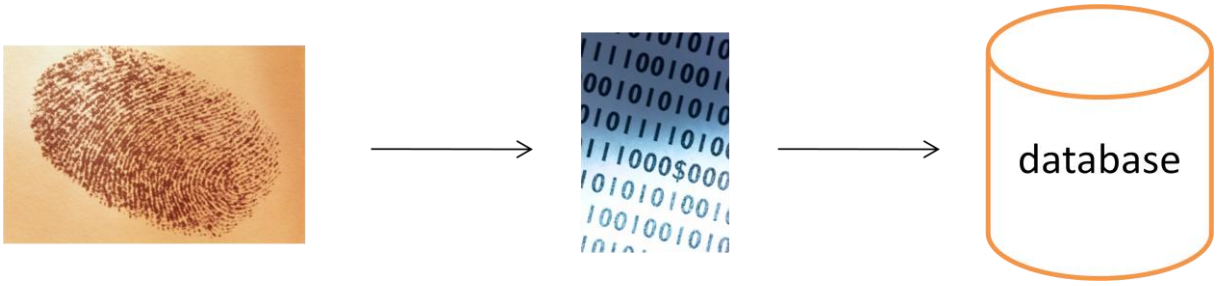

Image 4, Main idea of Encrypted Sketches

Before the biometric templates can be stored there has to be a setup. This has to be done by the service provider. He generates a Goldwasser-Micali key pair (*pk*, *sk*) and publishes *pk*; the public key.
In the enrollment phase when a user U_i wants to register his biometric template b_i to the service provider the following computations are done. After getting the biometric template, the service provider computes $P = SS_c(b_i) = c \oplus b_i$ and $[P]$ is stored at the database. $[\,.\,]$ denotes a related encryption Enc(. , *pk*). Also $H(c)$ is computed and then stored by the service provider. *C* is a randomly chosen codeword and *H* is a cryptographic hash function.
If a user wants to authenticate itself to the service provider, first a new biometric template b' is captured and $[b']$ is sent to the database. Next the database computes $[P] \times [b'] = [c \oplus b_i \oplus b'] = Z$ and sends *Z* back to the service provider. Then the service provider decrypts *Z* with his private key *sk* and also decodes the

output of $c \oplus b_i \oplus b'$. This has to be done to obtain a codeword c'. Last thing to do is to check, if $H(c') = H(c)$.
Because of the homomorphic property of Goldwasser-Micali scheme the database as well as the service provider never gets any information from the biometric template, because it stays encrypted all the time. Also you cannot obtain information about the codeword, since all computations are made in an encrypted way.

3. Private Information Retrieval Protocols (PIR)

Private Information Retrieval Protocols permits a user to recover data or an item, here a biometric template, from a server in possession of a database. But it is not revealed which item or biometric template the user currently asks for.

3.1 PIR

Assuming that a database is procured with M bits X = x_1, …, x_M. This database is secure if the PIR protocol fulfills the properties below.

- **Soundness:** If the user and the database succeed the protocol only the requested bit is the result of the query.
- **Request Privacy:** For all $X \in \{0,1\}^M \; for \; 1 \leq i,j \leq M$ and any used algorithm by the database, it is not possible for the database to differentiate between the request of index i and j; with respect to a non-negligible probability.

There are also other PIR constructions possible like a Symmetric PIR where the user only learn the information that he had requested. Another possibility is block-based PIR which works on block of bits.

3.2 Lipmaa's PIR

This authentication protocol uses Lipmaa's PIR instead of one of the former described PIR's, because it has one of the best known communication complexities. The main idea is that the database *S* is seen as a multidimensional array and the entries are associated to a vector of index. The size of *S* is defined as follows: $\prod_{j=1}^{\lambda} l_j$ where the integers l_j are the size of *S*. For $i = (i_1, \dots, i_\lambda)$ with $i_j \in \mathbb{Z}_{l_j} \; for \; j = 1, \dots, \lambda$, then S is the following:

$$S[i] = S\left[i_1 \prod_{j=2}^{\lambda} l_j + i_2 \prod_{j=3}^{\lambda} l_j + \cdots + i_{\lambda-1} l_\lambda + i_\lambda + 1 \right]$$

The idea of this PIR is to reduce the dimension of *S* successively through constructing a skimpier database in a recursive way until the last dimension is reached. This is done in the following way. Let S_0 = *S* as start and $(q_1, \dots, q_\lambda)$ the

request of some data. The answer is computed as follows: $S_1(i_2, \dots, i_\lambda)$ is the encryption of $S_0(q_1, i_2, \dots, i_\lambda)$. Now you have to loop this λ times until you get S_λ. This is a λ times encryption of $S_0(q_1, \dots, q_\lambda)$ and also is the answer of the request from the user.
The user can recover the requested element with successive decryptions, starting from $j = \lambda \; to \; j = 1$. The process of decryption is pictured below. At the end the result is the answer to the requested element.

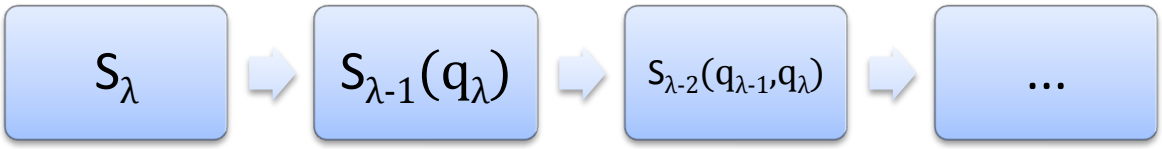

Image 5, decryption of the requested element

Request Privacy of Lipmaa's PIR is achieved through the semantic security of the Damgård-Jurik cryptosystem used to encode the request's index.

4. Authentication Protocol with encrypted Biometric Data

The authentication protocol consists of the following parts, depicted in the image below.

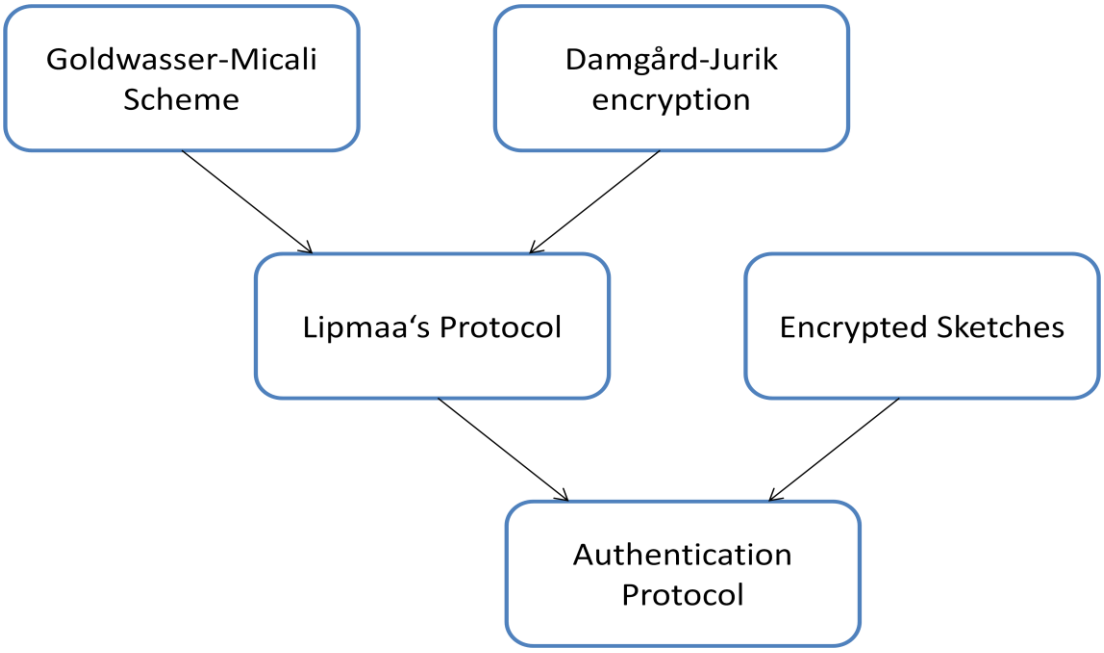

Image 6, Components of the authentication protocol

Identity privacy is achieved by small communication costs in contrast to previous construction and the whole data is protected during the entire process because it stays encrypted all the time.

4.1 The Protocol

For simplification the dimension of the database is set to $\lambda = 1$, but it is possible to do the following steps of the authentication protocol with any value for λ.
Here we utilize a double encryption consisting on the one hand of Goldwasser-Micali and on the other hand of Paillier Cryptosystem. In the following verification phase of the protocol $[.]$ identify a Goldwasser-Micali encryption and $[\![.]\!]$ an encryption via Paillier. There are further notations and information's to know before to fully understand the verification phase.

- The number of enrolled users in the database is M. So M is also the number of enrolled sketches in the database. Also the database owns the hash values for every secure sketch.
- The service provider has two pairs of keys. First the Goldwasser-Micali key pair (pk_{GM}, sk_{GM}) and second the key pair for the Paillier cryptosystem (pk_P, sk_P). Public keys are published and the private keys are stored in the Hardware Security Module.

If a user wants to authenticate himself to the service provider the following steps have to be done:

Step 1: New biometric template b' is encrypted by the sensor. This encryption is done by the Goldwasser-Micali scheme and the result is [b']

Step 2: The database gets a request from the client C. This request consists of the Paillier's ciphertext

$$[\![\delta_k^u]\!], k = 1, \dots, M, u = 0, \dots, l \; where \; \delta_k^u = [\pi_u(b')] \; if \; k = i \; and \; else \; 0$$

Step 3: Now the database computes the following:

$$[\![a_{i,u} \times [\pi_u(b')]]\!] = \prod_{k=1}^{M} [\![\delta_k^u]\!]^{a_k,u} \; for \; u = 0, \dots, l-1$$

and

$$[\![a_{i,l}]\!] = \prod_{k=1}^{M} [\![\delta_k^u]\!]^{a_k,l}$$

Step 4: After the computation is done, the database sends it to the service provider for u = 0, …,l.

Step 5: Next the Hardware Security Module (HSM) decrypts the received results first with via Paillier decryption algorithm and then with the Goldwass-

er-Micali decryption algorithm. Through this $H(c_i)$ is recover which is used in the next step.

Step 6: The HSM decodes $SS_c(b_i) \oplus b'$ to receive a codeword c'. Then it checks if $H(c') = H(c_i)$. If they are equal the authentication is accepted otherwise rejected.

Step 7: Finally the result is send to the service provider.

The above construction can be generalized, if you use a dimension $\lambda > 1$ in the Lipmaa PIR protocol. Furthermore the combination of Goldwasser-Micali and the used PIR is only possible if the group laws of Damgård-Jurik or the Paillier cryptosystem that are used in this authentication scheme, in general the underlying homomorphic encryption scheme, is compatible. Also one advantage of the used PIR is that it decreases the communication complexity in contrast to the previous version.

4.2 Security Analysis

To proof the security of this protocol we have to check three properties. The first is soundness which is defined as follows.

Definition 1: *A biometric-based authentication scheme is defined to be sound if it satisfies the following requirements: The service provider will accept an authentication request if the sensor client sends* (ID_i, b_i') *in an authentication request, where* b_i *and* b_i' *are matching data and* b_i *is the reference template registered for* ID_i*; and will reject it if they are non-matching data.*

(Bringer, et al., 2008)

The presented authentication protocol fulfills this definition, if the Secure Sketch is sound and the PIR protocol as well. This is guaranteed since the biometric data of the user and the sketches as well are always encrypted via a semantically secure encryption scheme.

For the next two properties we use an experiment called “attack game”. In this “game” an adversary generates a specific number of pairs of username and two associated biometric templates. Then a challenger chooses one of these biometric templates randomly and does the enrollment phase. After this the adversary tries to find out, which biometric template was chosen by the challenger via listening to a polynomial amount of verifications. The probability to learn or control which biometric template is used on the sensor side is negligible. So Identity Privacy and Transaction Anonymity are ensured.

Now the second one is Identity Privacy, which states that the database cannot find a relationship between a user and his biometric template. This is defined as follows.

Definition 2: *A biometric-based authentication scheme achieves identity privacy if* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *has only a negligible advantage in the following game, where the advantage is defined to be* $\left|Pr[e' = e] - \frac{1}{2}\right|$.

$$
\begin{aligned}
&Exp_{\mathcal{A}}^{Identity\ -Priacy}\\
&(i, ID_i, b_i^{(0)}, b_i^{(1)}, (ID_j, b_j)(j \neq i)) \leftarrow \mathcal{A}_1(1^l)\\
&b_i = b_i^{(e)} \xleftarrow{R} \left\{b_i^{(0)}, b_i^{(1)}\right\}\\
&\emptyset \leftarrow Enrollment\left(\left(ID_j, b_j\right)_j\right)\\
&e' \leftarrow \mathcal{A}_2(1^l)
\end{aligned}
$$

(Bringer, et al., 2008)

In this protocol the Identity Privacy is achieved only through the semantic security of the Goldwasser-Micali scheme under the QR assumption. Further it can be assumed that errors between two matching biometric templates b_i and b_i' of any User, where b_i is the template used at in the enrollment phase and b_i' the template used in the verification phase, are indistinguishable between all other possible errors that can occur. For a proof of this see (Bringer, et al., 2008).
Last one is Transaction Anonymity, which means that the database gets no information about which user is authentication himself and what is the result of the authentication.

Definition 3: *A biometric-based authentication protocol achieves transaction anonymity if a malicious database represented by an adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ *has only a negligible advantage in the following game, where the advantage is defined to be*
$\left|Pr[e' = e] - \frac{1}{2}\right|$.

$$
\begin{aligned}
&Exp_{\mathcal{A}}^{Transaction\ -Anoymity}\\
&\left(ID_j, b_j\right)(1 \leq j \leq N) \leftarrow \mathcal{A}_1(1^l)\\
&\emptyset \leftarrow Enrollment\left(\left(ID_j, b_j\right)_j\right)\\
&\{i_0, i_1\} \leftarrow \mathcal{A}_2(Challenger, Verification)\\
&i_e \xleftarrow{R} \{i_0, i_1\}\\
&\emptyset \leftarrow Verification(i_e)\\
&e' \leftarrow \mathcal{A}_3(Challenger, Verification)
\end{aligned}
$$

(Bringer, et al., 2008)

The Transaction Anonymity of the verification requests towards the database can directly concluded from the Request Privacy of the PIR protocol of Lipmaa that is used in the protocol. Against the service provider there is no transaction anonymity because it can learn the value of $H(c_i)$ for a specific authentication request of a

user U_i. Via this it is possible to track this specific user in future authentication requests. A countermeasure for this is to renew regularly the enrolled data, which means the encrypted sketch and the hash value that belongs to this. So the service provider is not able to track future authentication requests.

5. Conclusion

The goal of this authentication protocol is the same as the previous version of Bringer (Bringer, et al., 2007). The main difference between this and the previous protocol is that here only encrypted biometric data is used at each point in time. As a result of this, matching algorithms do not work anymore. So you have to use error correction procedures. This is possible through the approach of secure sketches. Also this authentication protocol can be built-in Lipmaa's PIR protocol.
But there are a lot of performance problems to take into consideration. First one is that the encryption by the Goldwasser-Micali scheme is done only bit by bit. In later authentication schemes it might be possible to use an encryption scheme which does the encryption not bit by bit but block wise or all in one. Another improvement can be done if computational aspects come to the fore instead of communication issues. Also the implementation of other PIR is possible.

6. Sources

Bringer Julien [et al.] An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication [Book Section] // ACISP 2007, LNCS 4586 / book auth. Pieprzyk J., Ghodosi H. and Dawson E.. - [s.l.] : Springer-Verlag, 2007.

Bringer Julien and Chabanne Hervé An Authentication Protocol with Encrypted Biometric Data [Book Section] // AFRICACRYPT 2008, LNCS 5023 / book auth. Vaudenay S.. - [s.l.] : Springer-Verlag, 2008.

7. Register of Illustrations