Fair Mutual Authentication

Jacek Cichoń[a], Krzysztof Majcher[b] and Mirosław Kutyłowski[c]
Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, Wrocław, Poland

Keywords: Fair Authentication, Authentication with Errors, Privacy Protection, GDPR, Markov Chain, Absorbing State, Rapid Mixing.

Abstract: We consider a fair authentication process where at each moment of the protocol execution each participant has almost the same certainty about the identity of the other participant. We combine this property with authentication with errors: each authentication bit may be replaced to the wrong value. Thereby, an observer attempting to derive the secret key(s) used for authentication in a cryptanalytic way has substantially harder job due to an unknown error pattern (learning secrets with errors). We show that the presented protocol satisfies subtle requirements of the GDPR Regulation of data minimization in case of failure.

1 INTRODUCTION

Assume that Alice and Bob wish to mutually authenticate themselves based on the fact that they are the only parties that know a secret K. There are many protocols based on cryptographic algorithms that enable both Alice and Bob to show that they hold a secret without exposing it. Some of them have zero-knowledge property – an observer has no advantage from listening to Alice and Bob: if there is an attack using information from interactions between Alice and Bob, then the adversary can run an analogous attack of almost the same complexity without eavesdropping communication between Alice and Bob.

Let us consider the following classical example of mutual authentication where Alice and Bob share a key K (this mechanism is used for Basic Access Control protocol for biometric passports (ICAO, 2015)):

Algorithm 1.

1. *Bob chooses a nonce r_B at random and sends it to Alice,*
2. *Alice chooses a nonce r_A at random and sends the ciphertext $Enc_K(r_A, r_B)$ to Bob,*
3. *Bob decrypts the ciphertext obtained from Alice and aborts if the plaintext does not contain r_B on the second position,*
4. *Bob returns $Enc_K(r_B, r_A)$ to Alice,*
5. *Alice decrypts the ciphertext and aborts if the plaintext is not (r_B, r_A).*

If a party reaches the end of the protocol execution without aborting, then it concerns the other party as authenticated. □

The main problem related to protocols of this kind is as follows:

Problem 2. *One of the parties (in this case Bob) reaches the state in which he is sure about identity of Alice, before Alice may judge whether she is interacting with Bob. At this moment Bob can interrupt the protocol prematurely or send invalid messages so that Alice finally will have no idea if she is interacting with Bob.*

Indeed, it the above example Bob is sure about Alice identity after step 3, while till this moment even the shared secret has not been used by Bob. Until step 5, Alice has no proof that she is interacting with Bob.

Problem 2 occurs also at everyday situations. For instance, if a call center of a bank is calling a bank's client, then both parties have to be authenticated. Typically this is done by exchanging some data that should be known by both parties. Whatever we do, one party says this information first. In electronic interaction the situation is difficult as well: first, it would be extremely difficult to create something that could be considered as really simultaneous exchange of messages. Second, in many cases for various reasons it is necessary to reduce the number of messages exchanged. In this case asymmetry of knowledge during protocol execution is inevitable.

[a] https://orcid.org/0000-0002-7742-3031
[b] https://orcid.org/0000-0003-2971-5571
[c] https://orcid.org/0000-0003-3192-2430

Cichoń, J., Majcher, K. and Kutyłowski, M.
Fair Mutual Authentication.
DOI: 10.5220/0010579507540759
In *Proceedings of the 18th International Conference on Security and Cryptography (SECRYPT 2021)*, pages 754-759
ISBN: 978-989-758-524-1

However, sometimes the technical reality is different. A prominent example are distance bounding protocols (see (Avoine et al., 2019)), where communicating parties rapidly exchange many short messages. In such a scenario one can design the following mutual authentication algorithm:

Algorithm 3 (Bitwise Exchange)**.** Assume that Alice and Bob share a secret K.

1. Alice chooses a nonce N_A at random, Bob chooses a nonce N_B at random,
2. Alice and Bob exchange N_A and N_B (in cleartext),
3. Alice and Bob compute $P_A = H(K, N_A, N_B, A)$ and $P_B = H(K, N_B, N_A, B)$ where H is a cryptographic hash function.
4. For $i = 1$ to k, the following steps are executed:
   - Alice sends a_i equal to the ith bit of P_A,
   - Bob checks that a_i is correct; if not, then he aborts the protocol execution,
   - Bob sends b_i, the ith bit of P_B,
   - Alice checks that b_i is correct; if not, then she aborts the protocol execution.

If a party reaches the end of execution without aborting, then its interlocutor is regarded as successfully authenticated. □

The important property of Algorithm 3 is that the number of correct authenticating bits revealed by Alice and Bob is almost the same at each step of the protocol execution. The difference is at most one.

1.0.1 Authentication with Errors

A frequent technique used for weak devices limited to lightweight cryptography is sending authentication data with a certain number of errors. A good example is the HB protocol and its variants: the underlying algebraic mechanisms of linear algebra are too weak from cryptographic point of view, but the messages exchanged contain a significant fraction of erroneous bits (see e.g. (Boureanu et al., 2017)). Thereby, an attacker is no more faced with a straightforward linear algebra problem, but with a kind of *learning secrets with errors* task. Learning with errors is one of the paradigms that results in designs not lomited to authentication – see e.g. (Bettaieb et al., 2018). This principle is also the foundation of the whole strain of cryptographic research based on the LPN problem and lattices (see e.g. lectures (Chi et al., 2015)).

Once the bits exchanged during the execution of Algorithm 3 are not necessarily correct in every case, then one could substantially reduce the requirements for the function H used. Like in case of HB-protocols, it is not necessarily a cryptographically strong hash function. This is a big advantage, since computing a hash value might be too complex for the simplest IoT devices.

One can propose the following version of Algorithm 3 where the authentication bits are partially false:

Algorithm 4 (Naive Algorithm)**.** The algorithm is the same as in case of Algorithm 3 apart from the bits exchange steps: For $i = 1$ to k, the following steps are executed:

- Alice sends a_i, where a_i equals the ith bit of P_A with probability p and its negation with probability $1 - p$,
- Bob checks that a_i is correct;
- Bob sends b_i, the ith bit of P_B, if a_i was correct, otherwise he sends its negation,
- Alice checks that b_i is correct iff a_i is correct; if not, then she aborts the protocol execution.

Bob accepts if the number of correct bits a_i exceeds a threshold $p \cdot k - \Delta$ where Δ is a parameter related to standard deviation. □

For Algorithm 4, an observer has no idea which bits are correct and correspond to the strings P_A and P_B. So we would like to claim that any brute force attack will have to guess the location of incorrect bits. Unfortunately, this is not true. Let α_i and β_i denote the ith bits sent, respectively, by Alice and Bob. Then of course $\alpha_i \oplus \beta_i = a_i \oplus b_i$, where $\oplus$ denotes the XOR operation. So the adversary may focus on finding a K such that the first k bits of $P_A \oplus P_B$ are the same as $\alpha_i \oplus \beta_i$ for $i = 1, \ldots, k$.

As we see we have to combine the following goals that to some extent are contradictory:

- a certain fraction of authentication bits sent by either party should be wrong (to confuse an adversary trying to retrieve the shared secret), while a sufficient majority must be correct (in order to enable reliable authentication),
- at each moment of protocol execution both authenticating parties should have almost the same level of certainty about the identity of the other party,
- the locations of erroneous bits on the side of Alice and on the side of Bob should be to some extent independent.

1.1 GDPR

There are many practical issues concerning discrepancy between technical reality and idealistic requirements of the the European GDPR regulation (The

European Parliament and the Council, 2016) on personal data protection. There are many cases where this dilemma has been revealed (see e.g. (Spindler and Schmechel, 2016), (Kutyłowski et al., 2020)), but pragmatic solutions to these problems are still missing.

Authentication protocols should be particularly carefully analyzed from the point of view of GDPR. Of course, electronic authentication is not performed directly by physical persons, but quite frequently the devices are attributed to their owners and indirectly provide data about them. It does not matter whether these data have any significance regarding information security of the holder, as the requirements of the GDPR regulation concern processing of personal data regardless of their significance.

If Alice executes a mutual authentication protocol with a party that declares to be Bob, then the permission to process the authentication data is given implicitly to Bob: executing the protocol is a form of consent of Alice, as it is *a clear affirmative act*. The addressee of this consent is definitely Bob, as long as the data sent by Alice depends on the identity of her interlocutor Bob. If we talk about privacy-by-design, almost no information should be delivered to a third party.

In case of algorithms such as Algorithm 1 no problem arises from the point of view of GDPR, when the protocol terminates in an accepting state on both sides. However, this cannot be claimed if, as described above, Bob interrupts the protocol execution or pretends not to be Bob by providing a false answer to the challenge of Alice. While Bob becomes sure about identity of Alice, the consent given by protocol execution has concerned a mutual authentication and not two one-way authentication protocols. From the legal point of view, a mutual authentication protocol would strictly follow the ideas of GDPR if the following properties are fulfilled:

Property 5 (GDPR Fully Compliant Mutual Authentication)**.** *A mutual authentication protocol executed by Alice and Bob should terminate on an accepting state on the side of Alice iff it terminates in an accepting state on the side of Bob. Moreover, if a protocol terminates in a state where Alice and Bob have only partial knowledge and cannot accept the interlocutor, then their degree of certainty should be comparable.*

Note that any deviation from the second requirement from Property 5 would lead to an asymmetry: the party having higher knowledge could interrupt the protocol and have an advantage over the other party. Property 5 should cover all cases of protocol execution: each participant may deviate from the protocol, including a malicious behavior.

1.2 Related Research

The problem discussed in this paper is closely related to fair exchange of information: in case of such a protocol Alice and Bob exchange some data, and neither of them should be advantaged to get the data before the other party. Research on these issues has been initiated decades ago. Already in 1980 it has been indicated (Even and Yacobi, 1980) that it is impossible to exchange data so that no party gets advantaged. However, it does not mean that one cannot create a protocol where at each moment of execution one party can be only slightly advantaged over the other party in the protocol – say by knowledge of one more information bit or even a fraction of it. In the protocols concerned the participants release their data gradually. For instance, the seminal paper (Blum, 1983) concerns exchange of private keys for two RSA numbers. The informations are exchanged bit by bit, interleaving the data sent by Alice and Bob.

While in the 80's the interest on fair exchange protocols have been more of a theoretical nature, these issues became extremely important due to the progress in electronic trade, where one party provides an electronic payment (e.g. with means of a cryptocurrency) and the other party provides a digital contents. In most practical business cases, the problem is resolved with a trusted arbiter. For example, an optimistic fair exchange protocol has been proposed in (Asokan et al., 1997). In this setting, once fairness is somehow broken, then a trusted party is involved and can resolve the issue at least pointing to the dishonest party. Somewhat related problems are solved by means of *smart contracts*, where the payment is unlocked by delivery of the purchased data and reflected in a blockchain. The application case that we are focusing on is of a different nature. Not only a third party would be cumbersome in most of technical settings, it would also create the problem of possible tracing the users, thereby breaking the fundamental principles of GDPR.

2 MARKOV FAIR MUTUAL AUTHENTICATION

For the rest of this paper we shall consider the following mutual authentication protocol that will ensure that each party shows at most one more correct bit than the other party at any moment of a protocol execution.

Algorithm 6 (Markov-Fair-MA)**.** Assume that Alice and Bob share a secret K. We describe this distributed

algorithm from the point of view of Alice. The interlocutor of Alice should follow analogous steps – however in a distributed environment Alice controls only herself and the interlocutor – allegedly Bob – may behave in an arbitrary way.

`Initialization:`

1. Alice chooses a nonce N_A at random and sends it in cleartext to Bob,
2. Alice receives N_B from Bob,
3. Alice computes $P_A = H(K,N,\text{“Alice”},\text{“Bob”})$ and $P_B = H(K,N,\text{“Bob”},\text{“Alice”})$, where $N = N_A \| N_B$ and H is a cryptographic hash function. Let a_i and b_i stand for, respectively, the ith bit of P_A and P_B.

`Main Part – Bit Exchange:` There are k rounds, at each round Alice sends one bit to Bob and receives one bit from Bob.
Let $\Delta_i = \delta^A_{i-1} - \delta^B_{i-1}$, where δ^A_j is the number of correct bits a_m sent by Alice before step j and δ^B_j is the number of correct bits b_m sent by Bob before step j.
Alice can be either in the `normal` state or in the `failure` state. The initial state is `normal`.
In round i the following steps are executed by Alice, if she is in the `normal` state:

- if $\Delta_i = -1$, then Alice sends a_i,
- if $\Delta_i = 0$ or $\Delta_i = 1$, then Alice sends a_i with probability p and $\neg a_i$ otherwise,
- if $\Delta_i > 1$, then Alice enters the `failure` state and sends a random bit.

Once Alice enters the `failure` state, then she remains in this state until the end of the protocol execution and at each remaining step sends a bit chosen at random.

`Decision:`
If Alice terminates the execution in a `normal` state, then she regards authentication as successful. □

Note 7. *It may not happen that Alice is in the `normal` state after step i and $\Delta_i < -1$. Indeed, note that $\Delta_0 = 0$ and at each step the value of Δ may change by at most 1. So if Δ drops below -1, then $\Delta_j = -1$ for step j immediately before. However, then Alice sends the correct value a_{j+1} at step $j+1$. If Bob at this moment sends the correct bit b_{j+1}, then $\Delta j+1 = -1$. If Bob sends an incorrect bit, then $\Delta_{j+1} = 0$, contradiction.*

Corollary 8. *If Alice in a normal state after step i, then $\Delta_i \in \{-1,0,1\}$.*

An execution of the protocol can be described by means of a Markov chain, with states corresponding to the value of $\Delta_i = -1,0,1$ and the `failure` state F. If Bob honestly follows the protocol, then the computation state from the point of view of Alice is described by the Markov chain $\mathcal{M}_C$ with the following state transition matrix

$$C = \begin{bmatrix} p & 1-p & 0 & 0 \\ (1-p)p & (1-p)^2+p^2 & (1-p)p & 0 \\ 0 & 1-p & p & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

(the rows 1, 2, 3, 4 correspond to, respectively, transition from the state -1, 0, 1 and F). The initial state of this Markov chain is 0. As long as Bob is following the protocol, then the state F is unreachable.

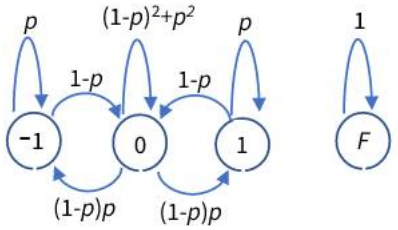

Figure 1: The Markov chain $\mathcal{M}_C$ describing the state transition when Alice is interacting with Bob honestly following the protocol.

The situation changes, if Alice is interacting with Eve impersonating Bob. As long as Eve does not know the shared secret K, then Eve cannot say which bit to be sent at round i will be regarded as correct by Alice. We make the following assumption that describes the situation for reasonable functions H:

Assumption 9 (forward security of H)**.** *Given the bits $a_1,\ldots,a_i$, and $b_1,\ldots,b_{i-1}$ Eve can guess the value of b_i with probability $\frac{1}{2}+\varepsilon$, where $\varepsilon \approx 0$ is negligible.*

In this situation, the state of the computation from the point of view of Alice can be described by the Markov chain $\mathcal{M}_F$ with the following state transition matrix

$$F = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1-p}{2} & \frac{1}{2} & \frac{p}{2} & 0 \\ 0 & \frac{1-p}{2} & \frac{1}{2} & \frac{p}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

It is important that the transitions of $\mathcal{M}_F$ are biased towards the state F, if $p > \frac{1}{2}$. For example, in the state 1 it is more likely to change to state F than to go to the state 0. So, from the point of view of the chain $\mathcal{M}_F$ the value of p should be as big as possible. On the other hand, for $\mathcal{M}_C$ the values of p close to 1 mean that it is hard to leave any state and therefore for the transmitted values α_i, β_i we have $\alpha_i \oplus \beta_i = a_i \oplus b_i$ with a high probability. So, the right choice for p is somewhere in the middle between $\frac{1}{2}$ and 1.

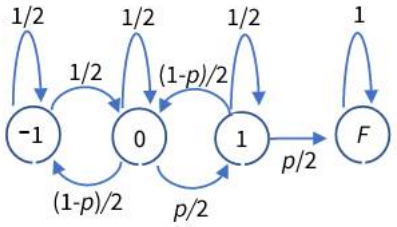

Figure 2: The Markov chain $\mathcal{M}_F$ describing the state transition when Alice is interacting with Eve unaware of the secret K shared by Alice and Bob.

3 BEHAVIOR OF Markov-Fair-MA

We will consider a general case of an arbitrary $p \in (0,1)$ used for the Markov-Fair-MA scheme. In Subsect. 3.3, we will check its properties for $p = \frac{2}{3}$.

3.1 Correct Executions

When confined to the states $-1, 0, 1$ the chain $\mathcal{M}_C$ has a stationary distribution π. Let D be transition matrix C after deleting the 4th column and the 4th row, i.e.

$$D = \begin{bmatrix} p & 1-p & 0 \\ (1-p)p & (1-p)^2+p^2 & (1-p)p \\ 0 & 1-p & p \end{bmatrix} \quad (3)$$

Then $\pi \cdot D = \pi$, so one can immediately derive that

$$\pi(-1) = \pi(1) = \frac{p}{2p+1}, \quad \pi(0) = \frac{1}{2p+1}. \quad (4)$$

The stationary distribution indicates what is the expected difference of the number of correct bits a_i and correct bits b_i. We see that the probabilities of the differences $-1, 0$ and 1 are not substantially different provided that neither $p \approx \frac{1}{2}$ nor $p \approx 1$. It shows that the process is not biased to some state and therefore the adversary should be confused about which bits are correct. However, the problem is that the process starts in the state 0 and the probability distribution of the states only converges to the stationary distribution. It is crucial to show that the distribution after round i converges quickly to the stationary distribution.

Let π_t denote the probability distribution of the state of $\mathcal{M}_C$ after t transitions of the chain, if the initial state is 0. A matrix D has three eigenvalues: 1, p and $p(2p-1)$, so using simple linear algebra we get:

Fact 10. *For any t we have*

$$\pi_t(j) = \begin{cases} \frac{p}{1+2p} - \frac{p}{1+2p}(p(2p-1))^t & \text{for } j = -1, 1 \\ \frac{1}{1+2p} + \frac{2p}{1+2p}(p(2p-1))^t & \text{for } j = 0 \end{cases} \quad (5)$$

From this equations we can directly derive the total variation distance

$$\|\pi - \pi_t\|_{TV} = \frac{1}{2} \sum_{j=-1,0,1} |\pi(j) - \pi_t(j)| \quad (6)$$

between the stationary distribution π and the distribution π_t:

Corollary 11.

$$||\pi - \pi_t||_{TV} = \frac{2p}{2p+1} |p(2p-1)|^t \quad (7)$$

As $|p(2p-1)| < 1$ for $p \in (0,1)$, we see that the total variation distance to the stationary distribution is decreasing exponentially. Note that the rate of convergence is nevertheless slow from the practical point of view, if p approaches 1. On the other hand, it is very high when p approaches $\frac{1}{2}$.

3.2 Execution with a Party Impersonating Bob

3.2.1 Convergence to the Absorbing State

The analysis of the Markov chain $\mathcal{M}_F$ is similar to the analysis of $\mathcal{M}_C$. The eigenvalues of the matrix F are

$$1, \tfrac{1}{2}, \tfrac{1}{2}(1-\sqrt{1-p^2}), \text{ and } \tfrac{1}{2}(1+\sqrt{1-p^2}). \quad (8)$$

This yields general formulas for the probability distribution π_t of the chain $\mathcal{M}_F$ after step t.

Fact 12.

$$\pi_t(j) = \begin{cases} \frac{1-p}{2\sqrt{1-p^2}}(b^t - a^t) & \text{if } j = -1, \\ \frac{p}{2\sqrt{1-p^2}}(b^t - a^t) & \text{if } j = 0, \\ \frac{1}{2}(b^t + a^t) & \text{if } j = 1, \\ 1 - \frac{1}{\sqrt{1-p^2}}(b^{t+1} - a^{t+1}) & \text{if } j = F, \end{cases} \quad (9)$$

where $a = \frac{1}{2}\left(1-\sqrt{1-p^2}\right)$ *and* $b = \frac{1}{2}\left(1+\sqrt{1-p^2}\right)$.

Let Y be a random variable denoting the first step when the process $\mathcal{M}_F$ reaches the absorbing state F. Noting that $\Pr[Y = t] = \Pr[Y \le t] - \Pr[Y \le t-1] = \pi_t(F) - \pi_{t-1}(F)$ and using equations 9 we get:

Fact 13. *For $t \ge 1$ we have*

$$\Pr[Y = t] = (b^t a - a^t b) \cdot \frac{1}{\sqrt{1-p^2}} \quad (10)$$

As a result of direct calculations we get:

Corollary 14. *The expected value and variance of Y have the following values*

$$\mathrm{E}[Y] = 4p^{-2} \quad (11)$$

$$\mathrm{Var}[Y] = 16p^{-4} - 12p^{-2} \quad (12)$$

Corollary 14 explains why too small values of p should be avoided. Namely, if $p \approx \frac{1}{2}$, then $\mathrm{E}[Y] \approx 16$. So it takes quite a long time until Alice will start to send purely random bits. In the meantime Alice may provide a partial proof of her identity.

3.2.2 Number of Visits in the State -1

The most critical moment from the point of information leakage when Alice interacts with an alleged Bob not knowing the shared key K, is the number of visits in the state -1 of $\mathcal{M}_F$. Indeed, in this case Alice must send the correct value of the corresponding bit a_t. In case of the states 0 and 1 there is only a bias to send the correct bit: so as the number of steps until the chain reaches the absorbing state F is small, it is hard to derive a meaningful statistical information.

Let Z be the random variable denoting the number of visits of the state -1 during an execution of the chain $\mathcal{M}_F$. Some computations involving equations 9 yield the following formulas:

Fact 15. $\mathrm{E}[Z] = \frac{2(1-p)}{p^2}$.

$$\mathrm{Var}[Z] = \frac{2(1-p)\left(3p^2-2p+2\right)}{p^4} \qquad (13)$$

Fact 13 shows that the expected number of visits in the state -1 is quite small for any reasonable choice of p. Also variance has relatively small values for $p \in (0.5, 1)$ (see Subsec. 3.3).

3.3 Example Choice: $p = \frac{2}{3}$

We have already noticed that neither the values of p close to $\frac{1}{2}$ nor the values values of p close to 1 is the right choice, so let us see what happens in the middle.

Honest Execution. In the case when Alice is Bob follows honestly the protocol, then the stationary distribution is given by vector $\pi = (\frac{2}{7}, \frac{3}{7}, \frac{2}{7})$. The total variation distance between distribution π_t of $\mathcal{M}_C$ and its stationary distribution is

$$\|\pi - \pi_t\|_{TV} = \tfrac{4}{7} \cdot \left(\tfrac{2}{9}\right)^t \ . \qquad (14)$$

So $\|\pi - \pi_5\|_{TV} \approx 0.0003$, $\|\pi - \pi_{10}\|_{TV} \approx 0.00000017$, $\|\pi - \pi_{32}\|_{TV} \approx 10^{-21}$.

Execution with Eve impersonating Bob. From Equations 11 and 12 we get that the expected time to reach the state F where Alice starts to send purely random bits is $\mathrm{E}[Y] = 9$ and $\mathrm{Var}[Y] = 54$, so the standard deviation is approximately 7.35.

By Fact 15 and Equation 13 we deduce that the number of visits of the state -1 before the process reaches the absorbing state F we have $\mathrm{E}[Z] = \frac{3}{2}$ and $\mathrm{Var}[Z] = \frac{27}{4}$, and the standard deviation is ≈ 2.6.

ACKNOWLEDGEMENTS

Authors would like to thanks Łukasz Krzywiecki for bringing attention to the problem discussed here.

REFERENCES

Asokan, N., Schunter, M., and Waidner, M. (1997). Optimistic protocols for fair exchange. In Graveman, R., Janson, P. A., Neuman, C., and Gong, L., editors, *Proc. 4th ACM Conference on Computer and Communications Security*, pages 7–17. ACM.

Avoine, G., Bingöl, M. A., Boureanu, I., Capkun, S., Hancke, G. P., Kardas, S., Kim, C. H., Lauradoux, C., Martin, B., Munilla, J., Peinado, A., Rasmussen, K. B., Singelée, D., Tchamkerten, A., Trujillo-Rasua, R., and Vaudenay, S. (2019). Security of distance-bounding: A survey. *ACM Comput. Surv.*, 51(5):94:1–94:33.

Bettaieb, S., Bidoux, L., Connan, Y., Gaborit, P., and Hauteville, A. (2018). The learning with rank errors problem and an application to symmetric authentication. In *2018 IEEE International Symposium on Information Theory, ISIT*, pages 2629–2633. IEEE.

Blum, M. (1983). How to exchange (secret) keys (extended abstract). In Johnson, D. S., Fagin, R., Fredman, M. L., Harel, D., Karp, R. M., Lynch, N. A., Papadimitriou, C. H., Rivest, R. L., Ruzzo, W. L., and Seiferas, J. I., editors, *Proc. 15th ACM Symposium on Theory of Computing*, pages 440–447. ACM.

Boureanu, I., Gérault, D., Lafourcade, P., and Onete, C. (2017). Breaking and fixing the HB+DB protocol. In Noubir, G., Conti, M., and Kasera, S. K., editors, *Proc. of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec*, pages 241–246. ACM.

Chi, D. P., Choi, J. W., Kim, J. S., and Kim, T. (2015). Lattice based cryptography for beginners. *IACR Cryptol. ePrint Arch.*, 2015:938.

Even, S. and Yacobi, Y. (1980). Relations among public key signature schemes. Technical Report 175, TECHNION.

ICAO (2015). Machine Readable Travel Documents - Part 11: Security Mechanism for MRTDs. Doc 9303.

Kutyłowski, M., Lauks-Dutka, A., and Yung, M. (2020). GDPR - challenges for reconciling legal rules with technical reality. In Chen, L., Li, N., Liang, K., and Schneider, S. A., editors, *Computer Security - 25th European Symposium on Research in Computer Security, Proc., Part I*, volume 12308 of *LNCS*, pages 736–755. Springer.

Spindler, G. and Schmechel, P. (2016). Personal data and encryption in the European General Data Protection Regulation. *JIPITEC*, 7(2):163–177.

The European Parliament and the Council (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/ec (General Data Protection Regulation). *Official Journal of the European Union*, 119(1).