

Probabilistic annotations for protocol models

— *Dedicated to Joshua Guttman* —

Dusko Pavlovic*

dusko@hawaii.edu

University of Hawaii, Honolulu HI, USA

Abstract

We describe how a probabilistic Hoare logic with localities can be used for reasoning about security. As a proof-of-concept, we analyze Vernam and El-Gamal cryptosystems, prove the security properties that they do satisfy, and disprove those that they do not. We also consider a version of the Muddy Children puzzle, where children’s trust and noise are taken into account.

1 Introduction

When it was first suggested that I should study security protocols, it was with a remark that the problem was largely solved and that I should simply look for a way to apply the solution to a particular protocol of interest, which happened to be one of the proposals for the IPsec suite. I found a paper that was circulating under the title ‘*How to solve any protocol problem*’ [8], and spent some time studying the methods of multi-party computation described in it. When I realized that I was not making any progress towards analyzing the IPsec protocol at hand, I went back and found out that the suggested solution of all protocol problems was not the multi-party computation, but strand spaces [10, 11, 15, 16, 17]. I drew the strand space bundles corresponding to the IPsec proposal the same afternoon.

Trying to save the science of security protocol design and analysis from its foretold demise, I spent a good part of the next 10 years looking for problems that could not be solved using the strand space model. Each time, I would then meet Joshua Guttman over dinner, usually at one of the Protocol eXchange meetings, and told him that there was this conceptual mismatch between his model and the reality, and he would then suggest how the problem of reality could be adjusted to match the strand spaces, and transformed towards a solution. On one

*Partially supported by NSF and AFOSR.

or two occasions when I was too far down the road towards different solutions, I avoided asking about the details.

But here is a record of something that definitely cannot be done by strand spaces. It has been clear from the outset that the strand space bundles can be annotated by Floyd-Hoare-style logical annotations [4, 6, 7], and that the various forms of dynamic and epistemic logics, worked out for reasoning about the preconditions, postconditions, and invariants of computations, can be elevated and generalized for reasoning about protocol security [2, 12] and about the higher-order properties of distributed systems and network interactions [13]. But what if we need to reason about the guessing chances, and have to go beyond the Dolev-Yao type of models [5]¹?

2 Crypto-logical systems

Towards a definition of a *crypto-logical system*, we begin from two basic data types: *states* \mathcal{S} and *predicates* \mathcal{P} . It is assumed that they are generated by a stratified set of algebraic operations, which allow us to write programs that lead to the states in \mathcal{S} , and to specify the resulting properties in \mathcal{P} . In particular, both \mathcal{S} and \mathcal{P} are built over the same algebra \mathcal{T} of terms, usually multisorted, assumed to contain enough variables, constants, and function symbols to specify keys, nonces, encryptions, decryptions, hashes, etc. These terms are computed, sent, and received by some actions that may be recorded in $q, s \dots \in \mathcal{S}$, while they may be compared, tested, and reasoned about in $\varphi, \psi \dots \in \mathcal{P}$.

2.1 Crude and overly general definition

Given a state space \mathcal{S} , an algebra of predicates \mathcal{P} , a set of agents \mathcal{A} , and a lattice of observations \mathcal{O} , a *crypto-logical system* is defined by the following data:

- a family of *semantic* maps

$$\mathcal{P} \times \mathcal{S} \times \mathcal{P} \xrightarrow{-\{-\}_A-} \mathcal{O}$$

indexed over the agents $A \in \mathcal{A}$,

- a measure

$$\mathcal{S} \xrightarrow{\mu} \mathbb{R}_+$$

given with a decomposition of \mathcal{S} into a disjoint union $\mathcal{S} = \coprod_{i \in I} \mathcal{S}_i$ of unit sets \mathcal{S}_i , i.e. such that $\Pr(\mathcal{S}_i) = 1$ holds for each $i \in I$. (Each restriction \Pr_i of \Pr to \mathcal{S}_i is thus a probability measure.)

¹There are, of course, many ways to go beyond the Dolev-Yao models and formalize probabilistic and computational reasoning in cryptography. One of the reviewers suggests that Easycrypt [1] should be mentioned. The point here is, however, to try to extend by probabilities the usual Floyd-Hoare annotations, which naturally fit with strand spaces.

Remarks. The above definition is more general than will be needed in this paper. But it conveys the big picture and the general path.

First of all, we do not need an abstract lattice of observations \mathcal{O} , but will always take $\mathcal{O} = \{0, 1\}$, and work with the usual Hoare triples $\varphi \{q\}_A \psi$, which are simply the elements of a ternary relation over $\mathcal{P} \times \mathcal{S} \times \mathcal{P}$. The reason for the above formulation is that the probabilistic analysis below will suggest that the *probabilistic* Hoare triples, evaluated in $\mathcal{O} = [0, 1]$, are also of interest, and in fact simplify some aspects of the reasoning. This option should be kept in mind for future work.

The decomposition of the state space $\mathcal{S} = \coprod_{i \in I} \mathcal{S}_i$ allowing the decomposition of the measure μ into the probability measures μ_i will also not play a significant role. It is in principle needed in the examples in Sections 3 and 4, where the state spaces will be certain powers of the monoid $\Sigma = \{0, 1\}^*$ of bitstrings, decomposed into $\Sigma = \coprod_{n=0}^{\infty} \{0, 1\}^n$, with the uniform probability distribution over each finite component $\{0, 1\}^n$. But this is spelled out in many textbooks, and the decomposition would flood the notations by information that is inessential for this paper, and hide the aspects that are essential. So we reduce the measure $\mu : \mathcal{S} \rightarrow \mathbb{R}_+$ to the component probabilities $\Pr : \mathcal{S} \rightarrow [0, 1]$, omitting the indices as they are easily reconstructed in all cases.

Furthermore, a state q in the space \mathcal{S} may or may not contain a record of a particular computation, run, or process that led to it. Short of a better word, we stretch the word "state" to mean "a result of a computation" — whatever part of it we may choose to record. Sometimes it may be the whole history, even including the intermediary results; sometimes just the outcome. A consequence is that \mathcal{S} may be closed under the usual programming and process operations, or it may be structured by the recorded data alone. In the former case, the usual rules of the Hoare logic will apply. In the latter case, when the concrete computations are not reflected by modal operators in \mathcal{S} , the Hoare notation boils down to

$$q \models_A \psi \iff \top \{q\}_A \psi$$

On the other hand, when the preconditions do play an essential role, relying upon the Hoare logic tradition and intuition seems appropriate, and useful.

In any case, we always require that the semantic maps $\mathcal{P} \times \mathcal{S} \times \mathcal{P} \rightarrow \mathcal{O}$ preserve the lattice structure of \mathcal{P} , contravariantly in the first, precondition argument, and covariantly in the postcondition.

2.2 Information sets and preorders of states

We say that, for an agent A , a process q' *refines* a process q , or that it *contains more information than* q , and we write $q \sqsubseteq_A q'$, whenever q' satisfies, as far as A

can tell, all the requirements that q satisfies:

$$q \sqsubseteq_A q' \iff \forall \varphi \psi. \varphi\{q\}_A \psi \leq \varphi\{q'\}_A \psi$$

Two processes are indistinguishable for the agent A if they satisfy the same requirements

$$\begin{aligned} q \sim_A q' &\iff q \sqsubseteq_A q' \wedge q \sqsupseteq_A q' \\ &\iff \forall \varphi \psi. \varphi\{q\}_A \psi = \varphi\{q'\}_A \psi \end{aligned}$$

The \sim_A -equivalence classes are A 's *information sets*. The quotient $\mathcal{S}_A = \mathcal{S} / \sim_A$ is A 's information view. A 's information set at q is written $q_A \in \mathcal{S}_A$.

2.3 Refining the definition of crypto-logical systems

The data type \mathcal{P} of predicates is assumed to support the usual logical connectives, which make it into a lattice. Moreover, it is also closed under a family of modalities W_ι , indexed over some *subjective evaluations* $\iota \in \mathcal{J}[0, 1]$, which will be just numbers between 0 and 1 in the simple examples below, but need to be generalized for some more involved cryptographic constructions. Semantics of these logical operations is defined by the following conditions

$$(\varphi_1 \vee \varphi_2) \{q\}_A \psi \iff (\varphi_1\{q\}_A \psi) \wedge (\varphi_2\{q\}_A \psi) \quad (1)$$

$$\varphi\{q\}_A (\psi_1 \wedge \psi_2) \iff (\varphi\{q\}_A \psi_1) \wedge (\varphi\{q\}_A \psi_2) \quad (2)$$

$$\varphi\{q\}_A (W_\iota \psi) \iff \Pr(\varphi\{s\}_A \psi \mid s \sim_A q) \in \iota \quad (3)$$

The Hoare triples here are the standard ones, evaluated in $\mathcal{O} = \{0, 1\}$, as explained in Sec. 2.1. Clause (3) extends the standard Hoare logic for probabilistic reasoning. The idea is that

- A 's subjective probability that ψ holds after φ at q is equal to
- the objective probability that ψ holds after φ at a randomly chosen state $s \sim_A q$.

By definition, the conditional probability in the last clause unfolds to

$$\Pr(\varphi\{s\}_A \psi \mid s \sim_A q) = \frac{\Pr\{s \in \mathcal{S} \mid s \sim_A q \wedge \varphi\{s\}_A \psi\}}{\Pr\{s \in \mathcal{S} \mid s \sim_A q\}}$$

The subjective vs objective probability conundrum goes back to the earliest days of probability theory [3], and persists as a useful distinction even in cryptographic reasoning. The objective probability is a number, which can be obtained, e.g., by counting frequencies. An observer of a random process,

however, may only be able to estimate that a probability falls within a certain interval, or just in a set, measurable modulo computational indistinguishability. There are thus various generality levels at which the family $\mathcal{J}[0, 1]$ of subjective evaluations may need to be modeled. To capture the standard cryptographic definitions in Sec. 3, the subjective evaluations from $\mathcal{J}[0, 1]$ will need to be feasibly computable subintervals of $[0, 1]$. For the simple examples presented in Sec. , on the other hand, rational numbers will suffice.

2.4 Probability vs. knowledge

Note that the statement $W_1\psi$, saying that ψ is satisfied with probability 1,

$$\varphi\{q\}_A (W_1\psi) \iff \Pr\left(\varphi\{s\}_A \psi \mid s \underset{A}{\sim} q\right) = 1$$

can be viewed as a generalization of the knowledge modality $K\psi$ for A defined by

$$\varphi\{q\}_A (K\psi) \iff \forall s \in \mathcal{S}. s \underset{A}{\sim} q \Rightarrow \varphi\{s\}\psi$$

where the logical implication $s \underset{A}{\sim} q \Rightarrow \varphi\{s\}\psi$ is replaced by the stochastic implication

$$[s \underset{A}{\sim} q \Rightarrow \varphi\{s\}\psi] \iff \Pr\left(\varphi\{s\}_A \psi \mid s \underset{A}{\sim} q\right) = 1$$

Intuitively, this stochastic implication says that the implication is valid almost everywhere, i.e. everywhere except at a set of measure 0. While the usual semantics of knowledge tells that $K\psi$ is satisfied for A after φ at q if ψ is satisfied after φ at every $s \underset{A}{\sim} q$, the probabilistic knowledge $W_1\psi$ is satisfied after φ for almost all $s \underset{A}{\sim} q$, i.e. with a possible exception of a set of measure 0. For each A , the statements $K\psi$ and $W_1\psi$ are *almost everywhere* equivalent, i.e. they only differ at a set of states of measure 0. Since cryptographic proofs are not just up to sets of measure 0, but usually identify even the ensembles that are computationally indistinguishable² the knowledge modality should, for all cryptographic purposes, be identified with W_1 .

2.5 Global semantics

We say that a requirement is satisfied globally if some agent observes that it is satisfied

$$\varphi\{q\}\psi \iff \exists X. \varphi\{q\}_X \psi \tag{4}$$

In practice, crypto-logical systems are often given by

²Two ensembles are computationally indistinguishable when their differences cannot be detected by polynomially bounded computations, e.g. because they occur only superpolynomially far down the strings of digits of their probabilities.

- a *global semantics*

$$\mathcal{P} \times \mathcal{S} \times \mathcal{P} \xrightarrow{-\{-\}-} \mathcal{O}$$

- a family of *views*

$$\mathcal{S} \xrightarrow{(-)_A} \mathcal{S}$$

indexed by $A \in \mathcal{A}$ such that

$$(\forall X. q_X = q'_X) \iff q = q' \text{ and} \quad (5)$$

$$\varphi \{q\} \psi \iff \exists X. \varphi \{q_X\} \psi \quad (6)$$

Local semantics can then be defined by

$$\varphi \{q\}_A \psi \iff \varphi \{q_A\} \psi$$

Condition (6) implies that (4) recovers the global semantics. Condition (5) implies that $q \sim_A q' \iff q_A = q'_A$. In other words, since all $q' \sim_A q$ satisfy the same requirements $\varphi \{q'\}_A \psi$ if and only if q_A satisfies them, then q_A can be taken as the canonical representative of the information set $[q]_A \in \mathcal{S}_A$.

2.6 Knowledge of probability vs probability of knowledge

The *logical* interpretation of the probabilistic modality W_ι , proposed in (3), was stated over the observations in $\mathcal{O} = \{0, 1\}$. Allowing the observations to be evaluated in $\mathcal{O} = [0, 1]$, and replacing the logical equivalence in (1) and (2) by the equality or indistinguishability of probabilities, leads to the probabilistic interpretation of the knowledge modality

$$\varphi \{q\}_A (K\psi) = \Pr \left(\varphi \{s\} \psi \mid s \sim_A q \right)$$

and promotes W_ι into a *confidence* modality

$$\varphi \{q\}_A (W_\iota \psi) = \Pr \left(\Pr \left(\varphi \{s\} \psi \mid s \sim_A q \right) \in \iota \right)$$

But this refined view has to be left for future work, as it requires first spelling out the standard view of familiar concepts, which barely fit in the rest of this paper.

3 Cryptographic definitions in crypto-logic

A *cryptosystem* consists of three agents, each executing a single probabilistic algorithm:

- key generation $\text{Gen} : \mathcal{R} \longrightarrow \mathcal{K} \times \mathcal{K}$,
- encryption $\text{Enc} : \mathcal{K} \times \mathcal{R} \times \mathcal{M} \longrightarrow \mathcal{C}$, and
- decryption $\text{Dec} : \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$,

such that

$$\text{Dec}(\bar{k}, \text{Enc}(k, x, m)) = m$$

where $\langle k, \bar{k} \rangle = \text{Gen}(y)$ for some $y \in \mathcal{R}$. Here \mathcal{R} represents the data type of random seeds, \mathcal{K} is the datatype of keys, \mathcal{M} the datatype of plaintext messages, and \mathcal{C} the ciphertexts. All datatypes are assumed to be finite, although unfeasibly large, so that it is sometimes convenient to assume that they are countably infinite. Each of them is given with a frequency measure

$$\text{Pr} : \mathcal{X} \longrightarrow [0, 1]$$

When no confusion seems likely, we shall denote a random variable sampling from \mathcal{X} also by \mathcal{X} , and write $\text{Pr}(x \in \mathcal{X})$ where most probability theory textbooks would write $\text{Pr}(\mathcal{X} = x)$.

Besides the principals of the cryptosystem, a definition of a security property that it may satisfy involves an attacker Att , which may operate any number of algorithms.

Remark. The notion of an *algorithm* is used here in the broadest sense, accommodating the various notions of computation. While the *computational* notions of security are defined assuming Probabilistic Polynomial-time Turing (PPT) machine as the standard model of computation, the *information-theoretic* security is defined over a notion of computation which boils down to mere *guessing* (of a message, a key, etc.), according to given frequency distributions. We begin with an information-theoretic definition.

Definition 3.1. A cryptosystem is perfectly secure if Attacker's chance to guess a message m at a state C , when he is given a ciphertext $c = E(k, x, m)$ is the same as his chance to guess that message at a state O , where he is not given any data, and can just randomly sample the space \mathcal{M} of messages:

$$C \models \mathsf{W}_t(m \in \mathcal{M}) \iff O \models \mathsf{W}_t(m \in \mathcal{M}) \quad (\text{IT-SEC})$$

Definition 3.2. Semantic (or chosen plaintext) security of a cryptosystem is tested by the following protocol:

- the Attacker computes (or randomly selects) two messages, m_0 and m_1 , and sends them to the Encryption oracle;
- the Encryption oracle tosses a coin, i.e. randomly selects a bit b , and a seed $x \in \mathcal{R}$, computes the ciphertext $c = E(k, x, m_b)$, and sends it to the Attacker.

The cryptosystem is semantically secure if Attacker's chance to compute (or to guess) the bit b at the final state C , when c is known to him, is not greater than his chance to guess b at the initial state O , without any data, i.e.

$$C \models W_t(b=1) \iff O \models W_t(b=1) \quad (\text{IND-CPA})$$

Definition 3.3. Adaptive (or chosen ciphertext) security of a cryptosystem is tested by the following protocol:

- the Attacker computes (or randomly selects) two messages, m_0 and m_1 , and sends them to the Encryption oracle;
- the Encryption oracle tosses a coin, i.e. randomly selects a bit b , and a seed $x \in \mathcal{R}$, computes the ciphertext $c = E(k, x, m_b)$, and sends it to the Attacker;
- the Attacker is then allowed to consult the Decryption oracle, to obtain the decryption $d = D(\bar{k}, c')$, of a chosen piece if ciphertext c' is feasibly constructed from m_0, m_1 and c , but differs from c , i.e. $c' \neq c$.

The cryptosystem is adaptive secure if Attacker's chance to compute (or to guess) the bit b at the final state C , when the ciphertext c and the decryption d are known to him, is not greater than his chance to guess b at the initial state O , without any data, i.e.

$$C \models W_t(b=1) \iff O \models W_t(b=1) \quad (\text{IND-CCA})$$

Remark. Varying the notion of computation in the above definition results in different notions of security. If the notion of computation is reduced to guessing, i.e. if the Attacker can only randomly choose m_0 and m_1 , and only randomly guess b , but possibly following a probability distribution skewed by the knowledge of c , then we get a weaker notion of security than the one where the Attacker can perform more structured computation, e.g. of a Probabilistic Polynomial-Time Turing Machine (PPT).

4 Examples of reasoning in crypto-logic

4.1 Security of the Vernam cryptosystem

In the Vernam cryptosystem, we take

$$\begin{aligned} \mathcal{K} &= \{0, 1\}^\ell \\ \mathcal{M} &= \mathcal{K}^j \\ \mathcal{C} &= \mathcal{M} \\ \mathcal{R} &= 1 \end{aligned}$$

and then define

$$E(k, m) = D(k, m) = k^j \oplus m$$

where \oplus is the *exclusive or* operation, and k^j is the j -tuple concatenation of a key k . We assume that the messages have a fixed number of blocks j just to avoid inessential notational details. The probability distributions over \mathcal{K} and over \mathcal{M} are given, and they determine

$$\Pr(c \in \mathcal{C}) = \sum_{k^j \oplus m = c} \Pr(m \in \mathcal{M}) \cdot \Pr(k \in \mathcal{K})$$

The Vernam cryptosystem is called *one-time pad* when $j = 1$, i.e. when a key is used to encrypt just one block.

Proposition 4.1. *One-time pad is perfectly secure. The Vernam cryptosystem is not perfectly secure for $j \geq 2$.*

Proof. To model the (IT-SEC) testing of the Vernam cryptosystem, we use as the states in \mathcal{S} the substrings of the triples $\langle k, m, c \rangle \in \mathcal{K} \times \mathcal{M} \times \mathcal{C}$, subject to the constraint that $c = k^j \oplus m$. Each state can be construed as the record of an encryption session, where the key k is first generated and sent from Gen to Enc , then the message m is chosen and encrypted by Enc into $c = k \oplus m$, and finally, the ciphertext c is sent to Dec and Att .

For each agent $X \in \{\text{Gen}, \text{Enc}, \text{Dec}, \text{Att}\}$ we define the view function $\mathcal{S} \xrightarrow{(-)_X} \mathcal{S}$ to be

$$\begin{aligned} \langle k, m, c \rangle_{\text{Gen}} &= \langle k \rangle \\ \langle k, m, c \rangle_{\text{Enc}} &= \langle k, m, c \rangle \\ \langle k, m, c \rangle_{\text{Dec}} &= \langle k, c \rangle \\ \langle k, m, c \rangle_{\text{Att}} &= \langle c \rangle \end{aligned}$$

The data type \mathcal{P} of predicates is generated from the formulas of binary arithmetic, extended with the probabilistic modalities W_ι .

We define semantics by stipulating that $\varphi \{q\}_X \psi$ is satisfied whenever the implication $\varphi(q_X) \Rightarrow \psi(q_X)$ is provable in binary arithmetic and elementary probability theory, starting from the given distributions $\Pr_{\mathcal{M}}$, and $\Pr_{\mathcal{K}}$.

Towards a proof of (IT-SEC) property for $j = 1$, first note that

$$\begin{aligned} \langle \rangle \models W_a(m \in \mathcal{M}) &\iff \Pr(m \in \mathcal{M}) = a \\ \langle c \rangle \models W_b(m \in \mathcal{M}) &\iff \Pr(m \in \mathcal{M} \mid c \in \mathcal{C}) = b \end{aligned}$$

On the other hand,

$$\begin{aligned} \Pr(m \in \mathcal{M} \mid c \in \mathcal{C}) &= \frac{\Pr(c \in \mathcal{C} \mid m \in \mathcal{M}) \cdot \Pr(m \in \mathcal{M})}{\Pr(c \in \mathcal{C})} \\ &= \Pr(m \in \mathcal{M}) \end{aligned}$$

holds because

$$\begin{aligned}\Pr(c \in \mathcal{C} \mid m \in \mathcal{M}) &= \Pr(c = k \oplus m \in \mathcal{C} \mid m \in \mathcal{M}) \\ &= \Pr(k = c \oplus m \in \mathcal{K} \mid m \in \mathcal{M}) \\ &= \Pr(k \in \mathcal{K})\end{aligned}$$

and

$$\begin{aligned}\Pr(c \in \mathcal{C}) &= \sum_{m \in \mathcal{M}} \Pr(c \in \mathcal{C} \mid m \in \mathcal{M}) \cdot \Pr(m \in \mathcal{M}) \\ &= \Pr(k \in \mathcal{K}) \sum_{m \in \mathcal{M}} \Pr(m \in \mathcal{M}) \\ &= \Pr(k \in \mathcal{K})\end{aligned}$$

It follows that $\langle \rangle \models W_a(m \in \mathcal{M})$ and $\langle c \rangle \models W_b(m \in \mathcal{M})$ are satisfied if and only if $a = b$.

For the Vernam cipher with $j \geq 2$, the probability $\Pr(c \in \mathcal{C} \mid m \in \mathcal{M})$ does not boil down to $\Pr(k \in \mathcal{K})$. Given $m = m_1 :: m_2 :: \dots :: m_j$, then c must be in the form $c = c_1 :: c_2 :: \dots :: c_j$ where $c_1 \oplus m_1 = c_2 \oplus m_2 = \dots = c_j \oplus m_j$ equals the key k . For $c \in \mathcal{C}$ which are not in that form, $\Pr(c \in \mathcal{C} \mid m \in \mathcal{M}) = 0$. For those that are, $\Pr(c \in \mathcal{C} \mid m \in \mathcal{M}) = \Pr(k \in \mathcal{K})$ remains valid. By a similar reasoning,

$$\Pr(m \in \mathcal{M} \mid c \in \mathcal{C}) = \begin{cases} \Pr(k \in \mathcal{K}) & \text{for } m_1 \oplus c_1 = \dots = m_j \oplus c_j \\ 0 & \text{otherwise} \end{cases}$$

This shows that the Vernam cryptosystem does not satisfy (IT-SEC) for $j \geq 2$. \square

Proposition 4.2. *If a Vernam cryptosystem is used to encrypt even one bit more than one block, then it is not semantically (IND-CPA) secure, i.e. it can be broken by a chosen-plaintext attack.*

Remark. Note that Attacker's capability to choose a plaintext is computational, and not just stochastic: they can determine the structure of the messages m_0 and m_1 in the CPA-test, and not just rather than just randomly sample from some source.

Proof of Prop. 4.2. To model the Vernam cryptosystem where one bit more than one block is encrypted, we take

$$\mathcal{M} = \mathcal{C} = \mathcal{K} \times \{0, 1\}$$

To model the (IND-CPA) testing of this cryptosystem, we use as the states in \mathcal{S} the substrings of the triples $\langle k, m_0, m_1, b, c \rangle \in \mathcal{K} \times \mathcal{M}^2 \times \{0, 1\} \times \mathcal{C}$, subject to the constraint that $c = k' \oplus m$, where $k' = k :: k_0$ is the key k with the first bit

repeated at the end. Each state can be construed as the record of an encryption session, where the key k is first generated by Gen , and securely conveyed to Enc and Dec , while on the other side the messages m_0, m_1 are generated by Att and sent to Enc , who then chooses the bit b , computes the ciphertext $c = k^j \oplus m_b$ and sends it c to Dec and Att .

For each agent $X \in \{\text{Gen}, \text{Enc}, \text{Dec}, \text{Att}\}$ we define the view function $\mathcal{S} \xrightarrow{(-)_X} \mathcal{S}$ to be

$$\begin{aligned}\langle k, m_0, m_1, b, c \rangle_{\text{Gen}} &= \langle k \rangle \\ \langle k, m_0, m_1, b, c \rangle_{\text{Enc}} &= \langle k, m_0, m_1, b, c \rangle \\ \langle k, m_0, m_1, b, c \rangle_{\text{Dec}} &= \langle k, c \rangle \\ \langle k, m_0, m_1, b, c \rangle_{\text{Att}} &= \langle m_0, m_1, c \rangle\end{aligned}$$

The data type of predicates \mathcal{P} and the semantics of $\varphi \{q\}_X \psi$ are just like in the proof of the preceding proposition.

Towards a proof that (IND-CPA) not satisfied, we note that $\langle \rangle \models W_{\frac{1}{2}}(b = 1)$ holds, because³ $\Pr(b = 1) = \frac{1}{2}$.

On the other hand, we show that the attacker can construct the messages m_0 and m_1 in such a way that $\langle c \rangle \models W_1(b = 1)$ holds if and only if $c = k' \oplus m_1$, and otherwise $\langle c \rangle \models W_0(b = 1)$ holds. Either way, $\langle c \rangle \models W_{\frac{1}{2}}(b = 1)$ does not hold, which implies that

$$\left(\langle c \rangle \models W_{\frac{1}{2}}(b = 1) \right) \iff \left(\langle \rangle \models W_{\frac{1}{2}}(b = 1) \right)$$

Towards the counterexample for (IND-CPA), let

$$\begin{aligned}m_0 &= 0^\ell :: 0 \\ m_1 &= 0^\ell :: 1\end{aligned}$$

which gives

$$\begin{aligned}c_0 &= k_0 :: k_1 :: \dots :: k_0 \\ c_1 &= k_0 :: k_1 :: \dots :: \neg k_0\end{aligned}$$

and

$$\begin{aligned}\langle c_0 \rangle &\models W_0(b = 1) \\ \langle c_1 \rangle &\models W_1(b = 1)\end{aligned}$$

□

³We assume that the coin is fair. If it is biased, the argument goes through for any probability p instead of $\frac{1}{2}$, provided that $p \neq 0$ and $p \neq 1$.

4.2 El-Gamal

Let \mathbb{G} be a cyclic group⁴ of order n with a generator g . In other words, the elements of \mathbb{G} can be listed in the form $g, g^2, g^3, \dots, g^{n-1}, 1$. The types of the El-Gamal cryptosystem are taken to be

$$\begin{aligned}\mathcal{K} &= \mathbb{G} \times \mathbb{Z}_n \\ \mathcal{R} &= \mathbb{Z}_n \\ \mathcal{M} &= \mathbb{G} \\ \mathcal{C} &= \mathbb{G} \times \mathbb{G}\end{aligned}$$

The keys $\langle k, \bar{k} \rangle = \text{Gen}(a)$ are set to be

$$\begin{aligned}k &= g^a \\ \bar{k} &= a\end{aligned}$$

and the encryption and decryption functions are

$$\begin{aligned}E(k, r, m) &= \langle g^r, k^r \cdot m \rangle \\ D(\bar{k}, c) &= \frac{c_2}{c_1^{\bar{k}}}\end{aligned}$$

where $c = \langle c_1, c_2 \rangle$. This defines a cryptosystem because

$$D(\bar{k}, E(k, r, m)) = \frac{k^r \cdot m}{(g^r)^{\bar{k}}} = \frac{g^{a \cdot r} \cdot m}{g^{r \cdot a}} = m$$

Definition 4.3. The Diffie-Hellman decision is the predicate $\text{DHd} : \mathbb{G}^3 \rightarrow \{0, 1\}$ defined by

$$\text{DHd}(x, y, z) \iff \exists a, b \in \mathbb{Z}_n. x = g^a \wedge y = g^b \wedge z = g^{ab}$$

where we abbreviate $\text{DHd}(x, y, z) = 1$ to $\text{DHd}(x, y, z)$, and write $\neg \text{DHd}(x, y, z)$ when $\text{DHd}(x, y, z) = 0$. The Decision Diffie-Hellman problem concerns the guessing algorithms for the Diffie-Hellman decision, i.e. the feasible algorithms with random seeds. The problem is that an algorithm should do better than a coin flip, and output more than half true decisions for a given length of the seeds. Formally, this means that for all $a, b \in \mathbb{Z}_n$ a DHd algorithm should satisfy⁵

$$\Pr \left(\text{DHd} (g^a, g^b, g^{ab}) \right) > \frac{1}{2}$$

⁴Here we hide away some details. \mathbb{G} is usually taken to be a cyclic subgroup of the multiplicative group of a field \mathbb{Z}_p . But while the reader familiar with the system, or a student of any cryptography textbook, will have no trouble recovering the details swept under the carpet, carrying them around here would distract from the main idea.

⁵It is required that the chance of $\text{DHd}(g^a, g^b, g^{ab}) = 1$ is feasibly distinguishable from $\frac{1}{2}$, i.e. greater by a feasible function. It follows that the chance of $\text{DHd}(g^a, g^b, g^d) = 1$ for $d \neq ab$ is also significantly smaller than $\frac{1}{2}$ by a feasible function.

The Decision Diffie-Hellman (DDH) assumption is that the Diffie-Hellman problem has no solution, i.e. that no feasible algorithm for guessing the Diffie-Hellman decision can do better than the coin flip.

Proposition 4.4. *The El-Gamal cryptosystem is semantically secure if and only if the Decision Diffie-Hellman assumption is true.*

Proof. To model the (IND-CPA)-testing of the El-Gamal cryptosystem, i.e. choosing the plaintexts that will yield distinguishable ciphertexts, we use as the states in \mathcal{S} the substrings of the tuples

$$\langle \langle k, \bar{k} \rangle, r, m_0, m_1, b, c \rangle \in \mathcal{K} \times \mathcal{R} \times \mathcal{M}^2 \times \{0, 1\} \times \mathcal{C}$$

where $k = g^{\bar{k}}$ and $c = \langle g^r, k^r \cdot m_b \rangle$. Each state can be construed as the record of a testing session, where the keys \bar{k} and k are generated, the first one is sent from Gen to Enc , the second one is announced publicly; the messages m_0, m_1 are chosen and sent from Att to Enc , the bit b and the ciphertext c are generated and sent from Enc to Att and Dec .

For each agent $X \in \{\text{Gen}, \text{Enc}, \text{Dec}, \text{Att}\}$ we define the view function $\mathcal{S} \xrightarrow{(-)_X} \mathcal{S}$ to be

$$\begin{aligned} \langle k, \bar{k}, r, m_0, m_1, b, c \rangle_{\text{Gen}} &= \langle k, \bar{k} \rangle \\ \langle k, \bar{k}, r, m_0, m_1, b, c \rangle_{\text{Enc}} &= \langle k, r, m_b, c \rangle \\ \langle k, \bar{k}, r, m_0, m_1, b, c \rangle_{\text{Dec}} &= \langle \bar{k}, m_b, c \rangle \\ \langle k, \bar{k}, r, m_0, m_1, b, c \rangle_{\text{Att}} &= \langle k, m_0, m_1, c \rangle \end{aligned}$$

Suppose that for the El-Gamal El-Gamal cryptosystem holds

$$C \models W_\iota(b = 1) \iff O \models W_\iota(b = 1) \quad (\neg \text{IND-CPA})$$

Since for a fair coin (i.e. uniformly distributed) $b \in \{0, 1\}$ it is certainly true that $O \models W_{\frac{1}{2}}(b = 1)$. The assumption $(\neg \text{IND-CPA})$ thus means that there is an attack that makes $C \models W_{\frac{1}{2}}(b = 1)$ false. There are thus algorithms

$$\text{Att}_0 : \mathcal{M}^2 \quad \text{and} \quad \text{Att}_1 : \mathbb{G} \times \mathcal{M}^2 \times \mathcal{C} \longrightarrow \{0, 1\}$$

such that for $\text{Att}_0 = \langle m_0, m_1 \rangle$ and any $b \in \{0, 1\}$ holds

$$\Pr \left(\text{Att}_1(k, m_0, m_1, \langle g^r, k^r \cdot m_b \rangle) = b \right) > \frac{1}{2}$$

The Diffie-Hellman decision $\text{DHd}(x, y, z)$ can now be computed for any given x, y and z from \mathbb{G} as follows:

- Set and announce the public key to be $k = x$.
- Let Att_0 generate and send the messages m_0 , and m_1 .

- Pick any $b \in \{0, 1\}$ and announce $c = \langle y, z \cdot m_b \rangle$.
- Set $\text{DHd}(x, y, z) = 1$ if and only if Att_1 correctly guesses b .

In summary,

$$\text{DHd}(x, y, z) = \begin{cases} 1 & \text{if } \text{Att}_1(k, m_0, m_1, \langle y, z \cdot m_0 \rangle) = 0 \\ & \text{and } \text{Att}_1(k, m_0, m_1, \langle y, z \cdot m_1 \rangle) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (\neg \text{DDH})$$

The other way around, assuming $(\neg \text{DDH})$ with a Diffie-Hellman decision algorithm DHd significantly better than a coin flip, the attacker Att_0 may generate m_0 and m_1 randomly, since Att_1 can always use DHd to decide which of the messages has been encrypted

$$\text{Att}_1(k, m_0, m_1, \langle c_0, c_1 \rangle) = \begin{cases} b & \text{if } \text{DHd}\left(k, c_0, \frac{c_1}{m_b}\right) \\ \perp & \text{otherwise} \end{cases}$$

Checking that this yields $(\neg \text{IND-CPA})$ is straightforward. \square

Proposition 4.5. *The El-Gamal cryptosystem is not adaptively secure, i.e. it can be broken by a chosen ciphertext attack.*

Proof. To model the (IND-CCA) (chosen ciphertext) testing of the El-Gamal cryptosystem, we use as the states in \mathcal{S} the substrings of the tuples

$$\langle \langle k, \bar{k} \rangle, r, m_0, m_1, b, q, c, c', d \rangle \in \mathcal{K} \times \mathcal{R} \times \mathcal{M}^2 \times \{0, 1\} \times \mathcal{R} \times \mathcal{C}^2 \times \mathcal{M}$$

where $k = g^{\bar{k}}$ and $c = \langle g^r, k^r \cdot m_b \rangle$, $c' \neq c$, and $d = D(\bar{k}, c')$. The projections can be

$$\begin{aligned} \langle k, \bar{k}, r, m_0, m_1, b, q, c, c', d \rangle_{\text{Gen}} &= \langle k, \bar{k} \rangle \\ \langle k, \bar{k}, r, m_0, m_1, b, q, c, c', d \rangle_{\text{Enc}} &= \langle k, r, m_b, c \rangle \\ \langle k, \bar{k}, r, m_0, m_1, b, q, c, c', d \rangle_{\text{Dec}} &= \langle \bar{k}, c', d \rangle \\ \langle k, \bar{k}, r, m_0, m_1, b, q, c, c', d \rangle_{\text{Att}} &= \langle k, m_0, m_1, q, c, c', d \rangle \end{aligned}$$

In order to gain advantage in determining b , the Attacker just needs to generate $q \neq 1$, and for $c = \langle c_1, c_2 \rangle$ set $c' = \langle c_1, q \cdot c_2 \rangle$. Then $d = q \cdot m_b$, and b can be determined with certainty, by comparing $m_b = \frac{d}{q}$ with m_0 and m_1 . \square

4.3 Towards protocols for noisy muddy mistrustful children

In some cryptanalytic attacks, the Attacker is a distributed system, consisting of several processes which locally make different observations, and send messages

to each other. The Muddy Children Puzzle can be viewed as a rudimentary example of such a situation. An unknown bitstring $m \in \mathcal{M} = \{0, 1\}^\ell$ can be thought of as denoting which members of a group of ℓ children have a muddy forehead. The fact that each child only sees other children's foreheads, but not its own, corresponds to the fact that an Attacker may consist of ℓ observers Att_i , $i = 1, \dots, \ell$, and each Att_i sees the bits m_k for $k \neq i$ but does not see m_i .

In the usual version of the puzzle, the father tells the children that at least one of them has a muddy forehead, and asks each child whether it knows if its forehead is dirty. He asks them in rounds: after they all say "No", he asks them all again, and so on. Using their view of other childrens' foreheads, and hearing their answers, each child can at some point tell whether its forehead is dirty. It is assumed that each child is a perfect reasoner: it will prove everything that can be proved at that point in time. At each point in time, each child either knows with certainty whether his forehead is muddy or does not know it at all.

In the probabilistic version, each child is trying to estimate the probability that his forehead is muddy. Initially, having finished playing together, the children have an estimate of the distribution $p : n \rightarrow [0, 1]$, where p_k is the probability that exactly k of them have a dirty forehead. If a child sees k dirty foreheads, then it knows for sure that there are either k or $k+1$ dirty foreheads altogether. So the initial probability that its own forehead is dirty is $\frac{p_{k+1}}{p_k + p_{k+1}}$.

Like in the usual version, each child then proceeds to announce, in rounds, whether it knows the state of its forehead. Knowing each other, they all also have an estimate of the probability that the statement that each of them is making is false (for one reason or another).

In other words, the Attackers initially know the probability p_k that there are exactly k 1s in m . Then each Att_i is allowed to broadcast to all Atts a message, telling whether he knows m_i or not. These broadcasts continue in rounds. After a finite number of such broadcasts, all Atts can compute all of the bitstring m .

The reasoning that allows this is one of the motivating examples behind knowledge logics. Generalizing the knowledge modality into the probability modality allows refined reasoning, where unreliability of the Attacker's communications can be taken into account: their broadcast bits can be flipped, with a given probability. This probability can be thought of as a measure of noise, or of mistrust among the children.

To be continued

While gathering the references, in particular those that I missed during the years of missed Protocol eXchanges, I encountered reports about the extensions of strand spaces, bundles, and shapes that support quantitative and hybrid forms or reasoning about security [9, 14, 18]. The tradition of Joshua explaining to me how what I presented could be done using the strand space model is hoped

to be continued in the future.

References

- [1] Gilles Barthe, François Dupressoir, Benjamin Grégoire, César Kunz, Benedikt Schmidt, and Pierre-Yves Strub. Easycrypt: A tutorial. In *Foundations of security analysis and design vii*, pages 146–166. Springer, 2013.
- [2] Ilano Cervesato, Catherine Meadows, and Dusko Pavlovic. An encapsulated authentication logic for reasoning about key distribution protocols. In Joshua Guttman, editor, *Proceedings of CSFW 2005*, pages 48–61. IEEE, 2005.
- [3] Lorraine Daston. How probabilities came to be objective and subjective. *Historia Mathematica*, 21(3):330–344, 1994.
- [4] Anupam Datta, Ante Derek, John Mitchell, and Dusko Pavlovic. A derivation system and compositional logic for security protocols. *J. of Comp. Security*, 13:423–482, 2005.
- [5] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Trans. Inf. Theor.*, 29(2):198–208, September 1983.
- [6] Nancy Durgin, John Mitchell, and Dusko Pavlovic. A compositional logic for proving security properties of protocols. *J. of Comp. Security*, 11(4):677–721, 2004.
- [7] Nancy Durgin, John C. Mitchell, and Dusko Pavlovic. A compositional logic for protocol correctness. In Steve Schneider, editor, *Proceedings of CSFW 2001*, pages 241–255. IEEE, 2001.
- [8] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or: A completeness theorem for protocols with honest majority. In *Proceedings of STOC*, New York, NY, USA, 1987. Association for Computing Machinery.
- [9] Joshua D. Guttman. Shapes: Surveying crypto protocol runs. In Véronique Cortier and Steve Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*, volume 5 of *Cryptology and Information Security Series*, pages 222–257. IOS Press, 2011.
- [10] Joshua D. Guttman. State and progress in strand spaces: Proving Fair Exchange. *J. Autom. Reason.*, 48(2):159–195, 2012.
- [11] Joshua D. Guttman. Establishing and preserving protocol security goals. *J. Comput. Secur.*, 22(2):203–267, 2014.

- [12] Catherine Meadows and Dusko Pavlovic. Deriving, attacking and defending the GDOI protocol. In Peter Ryan, Pierangela Samarati, Dieter Gollmann, and Refik Molva, editors, *Proceedings of ESORICS 2004*, volume 3193 of *Lecture Notes in Computer Science*, pages 53–72. Springer Verlag, 2004.
- [13] Dusko Pavlovic and Catherine Meadows. Actor Network Procedures. In Ram Ramanujam and Srinivas Ramaswamy, editors, *Proceedings of International Conference on Distributed Computing and Internet Technologies 2012*, volume 7154 of *Lecture Notes in Computer Science*, pages 7–26. Springer Verlag, 2012. arxiv.org:1106.0706.
- [14] John D. Ramsdell, Daniel J. Dougherty, Joshua D. Guttman, and Paul D. Rowe. A hybrid analysis for security protocols with state. In Elvira Albert and Emil Sekerinski, editors, *Proceedings of 11th IFM: Integrated Formal Methods*, volume 8739 of *Lecture Notes in Computer Science*, pages 272–287. Springer, 2014.
- [15] F. Javier Thayer, Jonathan C. Herzog, and Joshua D. Guttman. Honest ideals on strand spaces. In *Proceedings of the 11th CSFW*, pages 66–77. IEEE Computer Society, 1998.
- [16] F. Javier Thayer, Jonathan C. Herzog, and Joshua D. Guttman. Mixed strand spaces. In *Proceedings of the 12th CSFW*, pages 72–82. IEEE Computer Society, 1999.
- [17] F. Javier Thayer, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: Proving security protocols correct. *J. Comput. Secur.*, 7(1):191–230, 1999.
- [18] F. Javier Thayer, Vipin Swarup, and Joshua D. Guttman. Metric strand spaces for locale authentication protocols. In M. Nishigaki et al, editor, *Proceedings of IFIPTM: Trust Management IV*, volume 321 of *IFIP Advances in Information and Communication Technology*, pages 79–94. Springer, 2010.