# Time series analysis of a confidentiality measure for an Encrypted search system

Alexander Towell *
Southern Illinois University-Edwardsville Math Department

April 30, 2021

**Abstract**

We perform a time series analysis of the confidentiality of an Encrypted search system with respec to a theoretical adversary who employs a known-plaintext attack on the search agents' Encrypted searches. We derive an estimator of the forecast distribution on the confidentiality measure, which may be used to inform policies such as when and how frequently a password change may be called for to maintain a minimum level of confidentiality.

## Contents



*atowell@siue.edu

# 1 Introduction

In *cloud computing*, it is tempting to store confidential data on (untrusted) cloud storage providers. However, a system administrator may be able to compromise the confidentiality of the data, threatening to prevent further adoption of cloud computing and electronic information retrieval in general.

The primary challenge is a trade-off problem between confidentiality and usability of the data stored on untrusted systems. *Encrypted Search* attempts to resolve this trade-off problem.

**Definition 1.** Encrypted Search *allows authorized search agents to investigate presence of specific search terms in a confidential target data set, such as a database of encrypted documents, while the contents, especially the meaning of the target data set and search terms, are hidden from any unauthorized personnel, including the system administrators of a cloud server.*

Essentially, *Encrypted Search* enables *oblivous search*. For instance, a user may search a confidential database stored on an untrusted remote system without other parties being able to determine the information need of the user searched (and on more sophisticated systems, they are also unable to determine which documents were relevant to the information need).

We denote any untrusted party that has full access to the untrusted remote system (where the confidential data is stored) the adversary.[1]

Despite the potential of *Encrypted Search*, *perfect* confidentiality is not theoretically possible. There are many ways confidentiality may be compromised. In this paper, we consider an adversary whose primary objective is to comprehend the confidential information needs of the search agents by analyzing their history of *encrypted* queries.

A simple measure of confidentiality is given by the proportion of queries the adversary is able to comprehend. We consider an adversary that employs a known-plaintext attack. However, since the confidentiality is a function of the history of queries, different histories will result in different levels of confidentiality over time.

We apply time series analysis to estimate the forecast distribution of the confidentiality measure. The forecast distribution provides the framework to estimate important security-related questions such as "what will our mean confidentiality six months from now be?"

We are interested in reasonably medium-term forecasts so that we can plan accordingly for the future, e.g., determining how frequently passwords should be reset to try to maintain a base level of confidentiality. Resetting them too frequently poses an independent set of problems, both from a security and usability standpoint, but failing to reset them when the risk of being compromised is too high defeats the central purpose of Encrypted search.

# 2 Encrypted search model

An information retrieval process begins when a *search agent* submits a *query* to an information system, where a query represents an *information need*. In response, the information system returns a

[1] A system administrator being a typical example.

set of relevant objects, such as *documents*, that satisfy the information need.

An *Encrypted Search* system may support many different kinds of queries, but we make the following simplifying assumption.

**Assumption 1.** *The query model is a* sequence-of-words.

The *adversary* is given by the following definition.

**Definition 2.** *The adversary is an untrusted agent that is able to observe the sequence of queries submitted by authorized search agent.*

The objective of the *Encrypted Search* system is to prevent the adversary from being able to comprehend the sequence of queries.

**Definition 3.** *A* hidden query *represents a confidential* information need *of an authorized search agent that is suppose to be incomprehensible to the adversary.*

The primary means by which *Encrypted Search* is enabled is by the use of cryptographic *trapdoors* as given by the following definition.

**Definition 4** (Trapdoor)**.** *Search agents map* plaintext *search keys to some cryptographic hash, denoted trapdoors.*

A trapdoor for a *plaintext* search key is necessary to allow an *untrusted Encrypted Search* system to look for the key in a corresponding confidential data set.

**Assumption 2.** *The* Encrypted Search *system uses a simple substitution cipher in which each search key is mapped to a unique trapdoor signature.*

The simple substitution cipher is denoted by

$$\mathrm{h}\colon \mathbb{X} \mapsto \mathbb{Y}\,, \tag{1}$$

where $\mathbb{X}$ is the set of *plaintext* search keys and $\mathbb{Y}$ is the set of *trapdoors*.

Since h is one-to-one, it is possible to *undo* the substitution cipher by some function denoted by

$$\mathrm{g}\colon \mathbb{Y} \mapsto \mathbb{X} \tag{2}$$

such that

$$x = \mathrm{g}(\mathrm{h}(x))$$

for every $x \in \mathbb{X}$.

In a time series, we have *one* entitty and $T$ measurements of it over time. A random time series is a sequence of random variables

$$\{Y_1, Y_2, \ldots, Y_T\},$$

typically denoted by $\{Y_t\}$ where $t$ is the time index, which can continuous or discrete. MoreThe time index is more appro

The measurements are $d$ dimensional and may be continuous, discrete, or some mixture. Frequently $d = 1$, which we denote a univariate time series, and the measurements are continuous.

We use upper-case to denote random variables and lower-case to denote realizations, thus $Y_t$ is a random value and $y_t$ is the realization of $Y_t$.

Thus, a realization of the time series $\{Y_t\}$ is given by denoted by $\{y_t\}$.

The time series of *plaintext* keyword searches submitted by the search agents is denoted by $\{x_t\}$. It is a $d = 1$ dimensional time series with a discrete time index and a discrete response.

The adversary may may directly observe $\{x_t\}$. Instead, he observes a time series of ciphers.

**Definition 5.** *The* cipher $\{c_t\}$ *is a discrete time and discrete response time series defined as*

$$c_t = \mathrm{h}(x_t).$$

Since the time series of *plaintext* is a priori non-deterministic, we model it as a random time series $\{X_t\}$ such that

$$\Pr(X_j = x_j | X_1 = x_1, \ldots, X_{j-1} = x_{j-1}). \tag{3}$$

That is to say, our plaintext language model does not incorporate other kinds of information, such as who the search agent is or what time of day it is. In section 7, we consider extensiosn of the model.

Since $\{c_t\}$ is a function of $\{x_t\}$, we may model the ciphers as a random time series $\{C_t\}$ where $C_j = \mathrm{h}(X_j)$.

## 3 Threat model: known-plaintext attack

The primary source of information is given by the observable time series of ciphers $\{c_t\}$, which is induced by the unobserved time series of plaintext $\{x_t\}$.

Other potential sources of information, such as side-channel information, is not included in the model we consider in this paper. See section 7 for some preliminary thoughts on this expanded topic.

In the threat model described in section 3, the adversary is interested in estimating $\{x_t\}$. However, the adversary is only able to observe $\{c_t\}$. Thus, the adversary's objective is to infer the plaintext from the ciphers using frequency analysis attacks, in particular a *known-plaintext* attack.

In a known-plaintext attack, the objective of the adversary is to learn how to *undo* the substitution cipher h with g.

**Assumption 3.** *The inverse substitition cipher* g *is not known to the adversary.*

A maximum likelihood estimator of g is given by

$$\hat{\mathrm{g}} = \arg\max_{\mathrm{g} \in G} \Pr(X_1 = \mathrm{g}(c_1)) \prod_{t=2}^{T} \Pr(X_t = \mathrm{g}(c_t) | X_{t-1} = \mathrm{g}(c_{t-1}), \ldots, X_1 = \mathrm{g}(c_1))$$

where $G$ is the set of all possible mapping functions from ciphers $\mathbb{Y}$ to plaintexts $\mathbb{X}$.

If two plaintexts $x, x' \in \mathbb{X}, x \neq x'$, may be exchanged without changing the probability of $\{x_t\}$, then they are *indistinguishable* and $\hat{\mathrm{g}}$ is *inconsistent*. However, the adversary does not need to be perfect for the confidentiality measure to be compromised. If some of the plaintexts are inexchangeable, then the adversary may learn *something* about $\{x_t\}$ by observing $\{c_t\}$.

The greater the uniformity of $\{X_t\}$ the greater the variance of $\hat{\mathrm{g}}$. At the limit of maximum uniformity, where every pair of plaintext is exchangeable, the adversary can learn nothing about $\{x_t\}$ by observing $\{c_t\}$. Natural languages have a high degree of non-uniformity and so the primary concern of the adversary is the divergence between the *true* distribution and the *known-plaintext* distribution.

**Assumption 4.** *The adversary knows some approximation of* $\{X_t\}$.

The *known-plaintext* distribution may be used to solve an approximation of the MLE $\hat{\mathrm{g}}$.

**Definition 6.** *In a* known-plaintext attack, *the adversary substitutes the unknown true distribution with the known-plaintext distribution and solves the MLE under this substituted distribution.*

# 4 Confidentiality measure

We are interested in measuring the degree of confidentiality as given by the following definition.

**Definition 7.** *Given a time series* $\{c_{t'}\}$, *the confidentiality measure is a time series* $\{\pi_t\}$ *defined as the fraction of ciphers in* $\{c_{t'}\}$ *that the adversary successfully maps to* plaintext *where* $t' = Nt$. *That is,*

$$\pi_t = \frac{\delta_t}{Nt}, \tag{4}$$

*where*

$$\delta_t = \sum_{t'=1}^{Nt} \left[\mathrm{g}(c_{t'}) = \hat{\mathrm{g}}(c_{t'})\right]. \tag{5}$$

Note that $N$ denotes the fact that we take one measurement of the confidentiality every time a multiple of $N$ ciphers are observed.

The measure $\pi_t$ can be understood as the marginal probability that the adversary is able to decode an incoming cipher to plaintext at around time $t$. However, far more revealingly, the adversary may go back through the history of ciphers and decode proportion $\pi_t$ to plaintext.

If we specify that $\pi^*$ is the minimum confidentiality measure we wish to maintain, then it is essential that we stop generating $\{c_t\}$ at or before time $T^*$ where

$$T^* = \arg\min_T \pi_T > \pi^*.$$

That is, we stop generating $\{c_{t'}\}$ before the amount of information in it is sufficient for the adversary to decode more than proportion $\pi^*$ of the data. We do not need to stop Encrypted search queries at time $T^*$, we only need to change the cipher, i.e., substitute the mapping function h that maps plaintexts to ciphers with some other mapping function, which is typically done by requiring users to change passwords periodically. This is where *forecasting* $\{\pi_t\}$ plays a central role.

## 4.1 Forecasting model

As a function of a *random time series* $\{C_{t'}\}$, we may model $\pi_t$ as being generated by the random time series $\{\Pi_t\}$. If $\pi_t$ is not known, i.e., $\Pi_t$ has not been observed, then $\Pi_t$ is a probability distribution on the measure at time $t$. If $\pi_1, \pi_2, \ldots, \pi_T$ is given, then $\Pi_{T+h|T}$ is a *conditional* distribution[2] known as the $h$-step forecast distribution at time $T$ whose expectation is denoted by $\pi_{T+h|T}$.

[2] $\Pi_{T+h}$ given $\Pi_1 = \pi_1, \ldots, \Pi_T = \pi_T$.

Our primary interest is in *forecasting* an observed time series $\{\pi_t\}$, e.g., if we observe $\{\pi_1, \pi_2, \ldots, \pi_T\}$, we wish to estimate the mean of the $h$-step forecast $\pi_{T+h|T}$. Since $\pi_{T+h|T}$ is not known, we seek an estimator $\hat{\pi}_{T+h|T}$.

## 5 Data description

The *accuracy* $\{\pi_t\}$ of the adversary is the single entity we are observing and we have $T$ measurements of it over logical time.

The confidentiality data $\{\pi_t\}$ depends upon two other time series, the plaintext (keyword searches) $\{x_t\}$ and the ciphers $\{c_t\}$, which Alex Towell generated in 2016 using the following steps:

1. The parameters of a Bigram language model were estimated from a large corpus of plaintext. (The source of the particular corpus used has been lost.)
2. The estimated Bigram language model was conditionally sampled from to generate plaintexts $\{x_t\}$.
3. Each plaintext $x_t$ was cryptographically hashed to a cipher $c_t = \mathrm{h}(x)$ to generate ciphers $\{c_t\}$.

Note that $\{x_t\}$ and $\{c_t\}$ are not the primary time series of interest in our analysis. Rather, our primary interest is in the confidentiality measures $\{\pi_t\}$. To generate this time series, the following steps were taken:

1. The function g that maps ciphers to plaintext is estimated after every $N = 50$ observations of the cipher time series using a MLE under a unigram language model (some information in the bigram model is not being used by the estimator, which reduces its efficiency) on a different corpus judged to be similiar to the one used to generate $\{x_t\}$. Thus, the unigram MLE of g at time $T$ is given by

   $$\hat{\mathrm{g}}_T = \arg\max_{\mathrm{g} \in G} \prod_{t=1}^{T} \hat{\Pr}(X_t = \mathrm{g}(c_t)).$$

   Note that $\hat{\mathrm{g}}_T$ is inconsistent since it does not converge in probability to g as a consequence of the adversary's estimation of $\Pr(X_t)$ with $\hat{\Pr}(X_t)$.

   This inconsistency was motivated out of a desire to be more realistic, since an adversary who is performing the known-plaintext attack cannot in practice know the underlying distribution of $\{x_t\}$ used to generate the keyword searches.
2. The confidentiality measure at time $t$, denoted by $\pi_t$, is computed using $\hat{\mathrm{g}}_t$.

## 6 Time series analysis of $\{\pi_t\}$

It seems clear that the adversary's accuracy at a particular time will be correlated with lagged (previous) values of its accuracy and the closer in time they are the more heavily correlated they will generally be (barring exceptions like seasonality).

We partition the data into a training set and a test set. We will not look at the test set until later when we evaluate the model. Here is a quick glimpse of the training set data:

```
## [1] 0.358159 0.351208 0.347271 0.346403 0.352666 0.350445
```

### 6.1 Visualization and stationary transformations

If the time series data can be transformed to meet the stationary conditions, then the ARIMA model for the (correlated) residuals is generally a reasonable choice. A stationary time series is given by:

1. The mean is not a function of time.
2. The variance is constant.
3. The autocorrelation is a function of lag rather than time.

A plot of the training partition of $\{\pi_t\}$ is shown in figure 1.

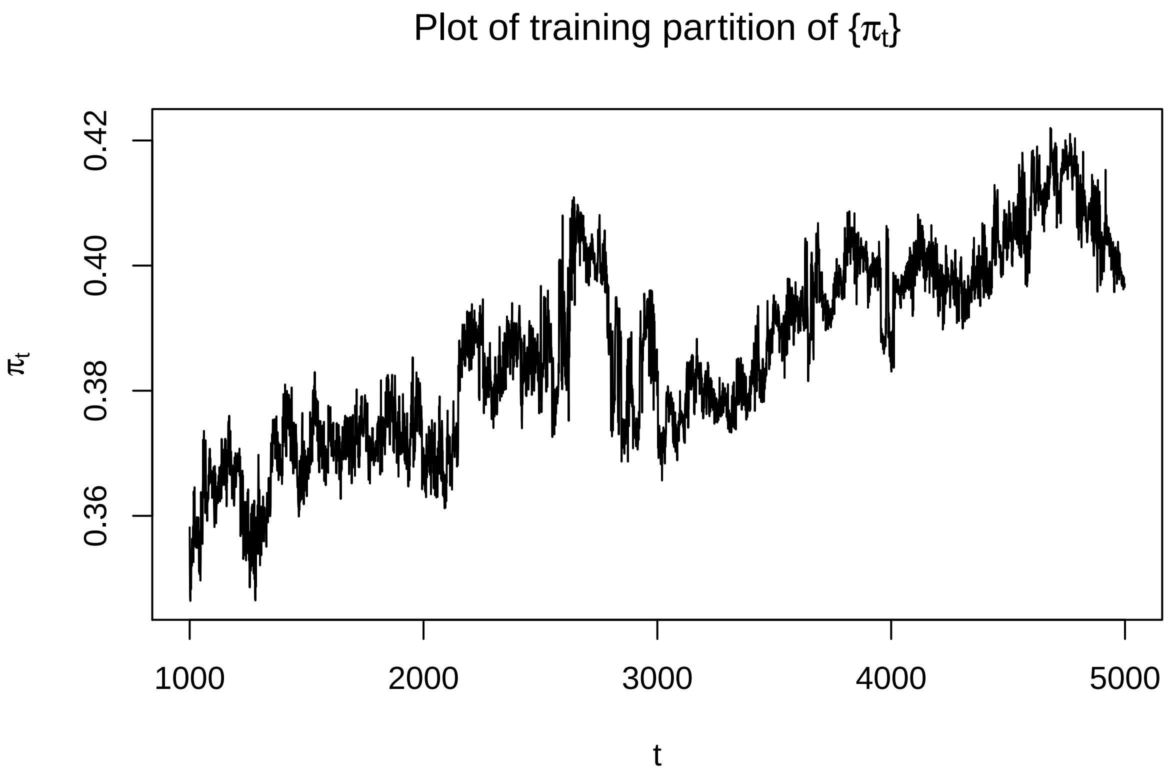


Figure 1: A non-stationary time series plot.

It appears non-stationary. A plot of the sample ACF and PACF are shown in figure 2.

We see that the ACF indicates that $\{\pi_t\}$ has significant autocorrelation. While this is clearly non-stationary, the variance seems constant and thus a transformation to make the variance more uniform, such as a log-transformation, seems unnecessary.

Since there is not necessarily an obvious pattern in the data, we will avoid the use of procedures like fitting a regression model (for detrending) and instead try some order of differencing. Differencing is a non-parametric approach that can often transform a non-stationary time series into a stationary one, where the $d$-th difference of $\{\pi_t\}$ is denoted by $\nabla^d\{\pi_t\}$. Moreover, since it is non-parametric, differencing has the added benefit of being able to dynamically respond to changes in the data, unlike with regression which treats the trend as deterministic.

In figure 3, we plot the differenced process $\nabla\{\pi_t\}$. We see that the trend has been removed, the values are centered around zero, and the variance is constant. We believe this may be stationary.

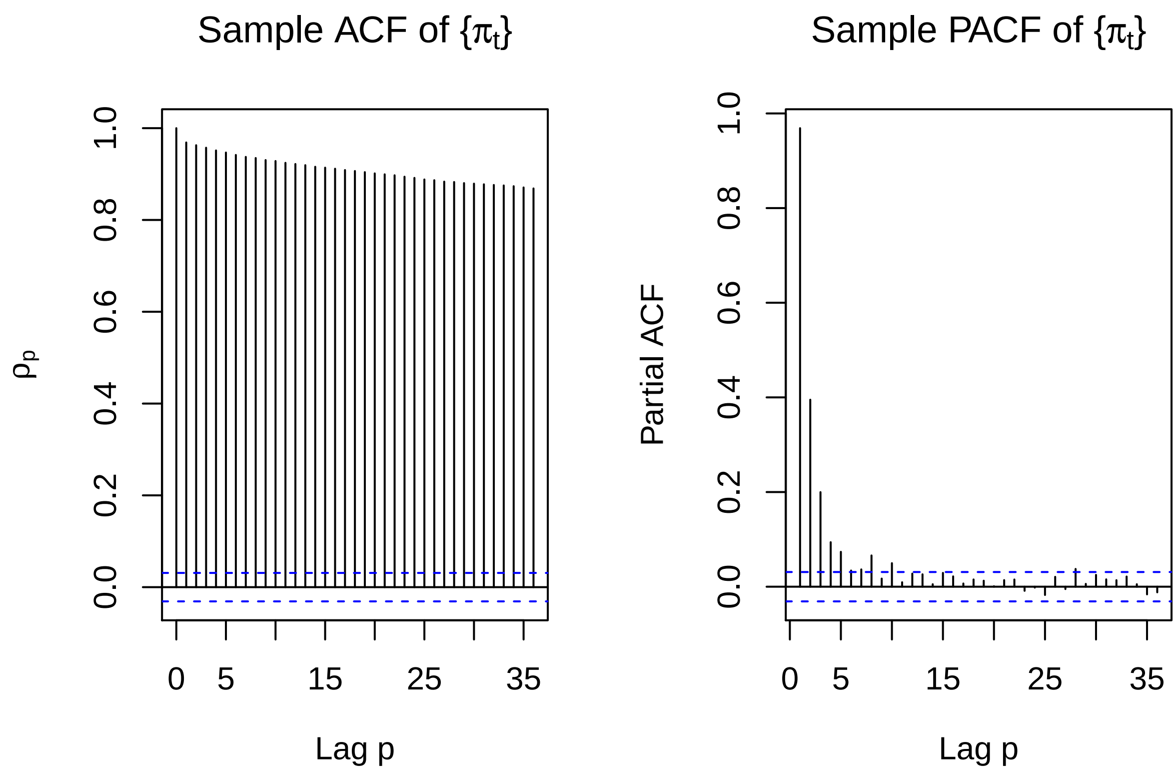


Figure 2: Highly correlated sample ACF and PACF.

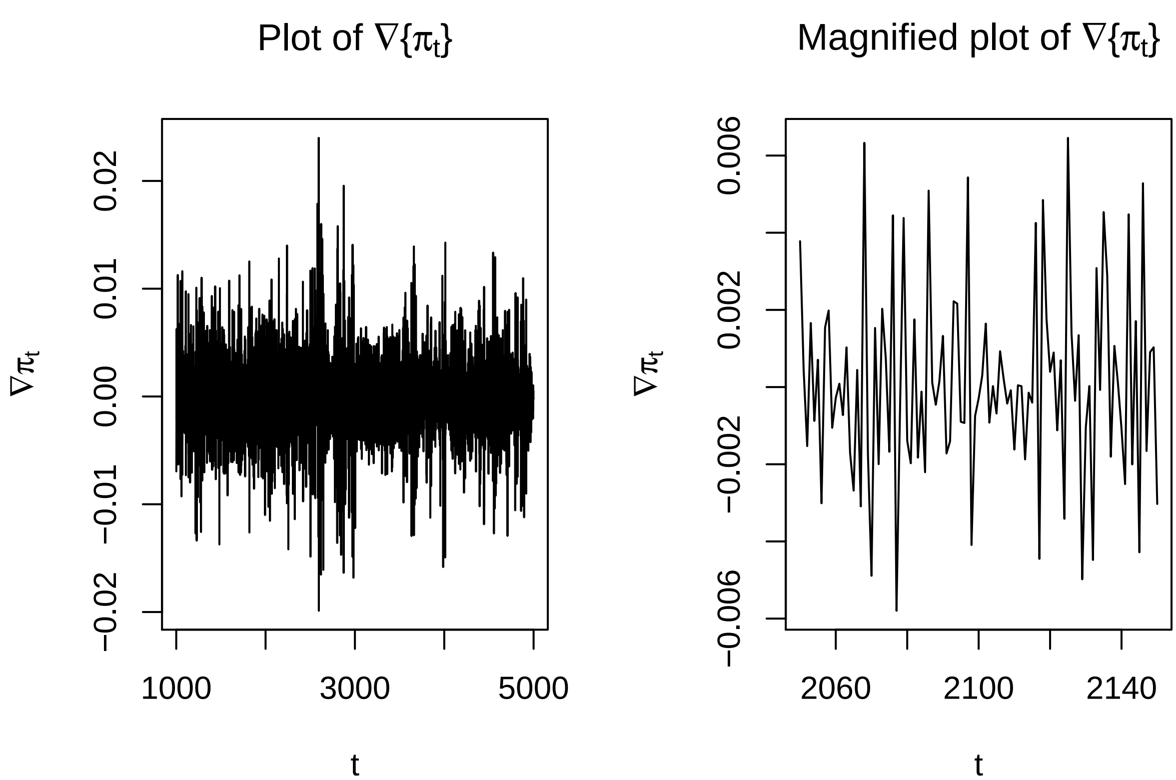


Figure 3: Time plots of $\nabla\{\pi_t\}$.

We perform the augmented Dickey-Fuller test[2] as a more objective measure.

```
## Augmented Dickey-Fuller Test
## Dickey-Fuller = -20.37, Lag order = 15, p-value = 0.01
## alternative hypothesis: stationary
```

The $p$-value of the Dickey-Fuller hypothesis test is less than 0.01, which we consider to be very strong evidence against the null hypothesis of non-stationary data. Bolstered by this test, we proceed with trying to find an ARIMA model for the residuals.

### 6.2 ARIMA model selection

There are perhaps three primary reasons we would want to infer a general model for $\{\pi_t\}$: prescription, description, and in our case, *prediction*.

Guided by the principle of *parsimony*, we have a bias for simpler models, i.e., Occam's razor. As justification for this bias, consider the following. Assume there is some unknown process $M$ that generated data $\{\pi_t\}$. If we have parametric model $M'$ with many degrees of freedom (dimension of parameter space), we may find parameters for it that fit it to $\{\pi_t\}$ with a very small sum of squared residuals.

However, if $M'$ is unnecessarily complex, it is unlikely to *generalize* very well, i.e., on new data $M$ and $M'$ may diverge significantly. In this case, we say that $M'$ is overfitted to the observed data $\{\pi_t\}$. Of course, if a simpler model cannot even sufficiently model the observed data $\{\pi_t\}$, it is hard to justify as an approximation of $M$. Thus, we have a variance-bias trade-off[1].

Since ARIMA models are parameterized by $p$, $d$, and $q$, which respectively specify the order of the autoregression component, the order of the difference, and the order of the moving average component, we have a bias for ARIMA models with relatively small $p$, $d$, and $q$. Note that there are many heuristics that model this bias, such as the Akaike information criterion (AIC), but we will decide upon a subset of candidate models that leans heavily on a more hands-on analysis.

A plot of sample ACF and PACF of the differenced time series $\nabla\{\pi_t\}$ is shown in figure 4.

Since the ACF cuts off after lag 1 and the PACF decays exponentially, we speculate that $\nabla\{\pi_t\}$ may be an MA(1) process. We use the EACF plot to try to help determine other possible orders of the ARIMA model:

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o o x x x o  o  o  o
## 1 x x o o o o x o o o o o  o  o  o
## 2 x x x o o o x o o o o o  o  o  o
## 3 x x x x o o o o o o o o  o  o  o
## 4 x x x x x o o o o o o o  o  o  o
## 5 x x x x x x o o o o o o  o  o  o
## 6 x x x x x o x o o o o o  o  o  o
## 7 x x x x x x x o o o o o  o  o  o
```

Ignoring the set small set of $x$'s above the main diagonal of zeros, we see that $\{\nabla\pi_t\}$ seems to be compatible with $\text{ARMA}(0,1)$,$\text{ARMA}(0,2)$, and $\text{ARMA}(1,2)$. We will analyze these three.

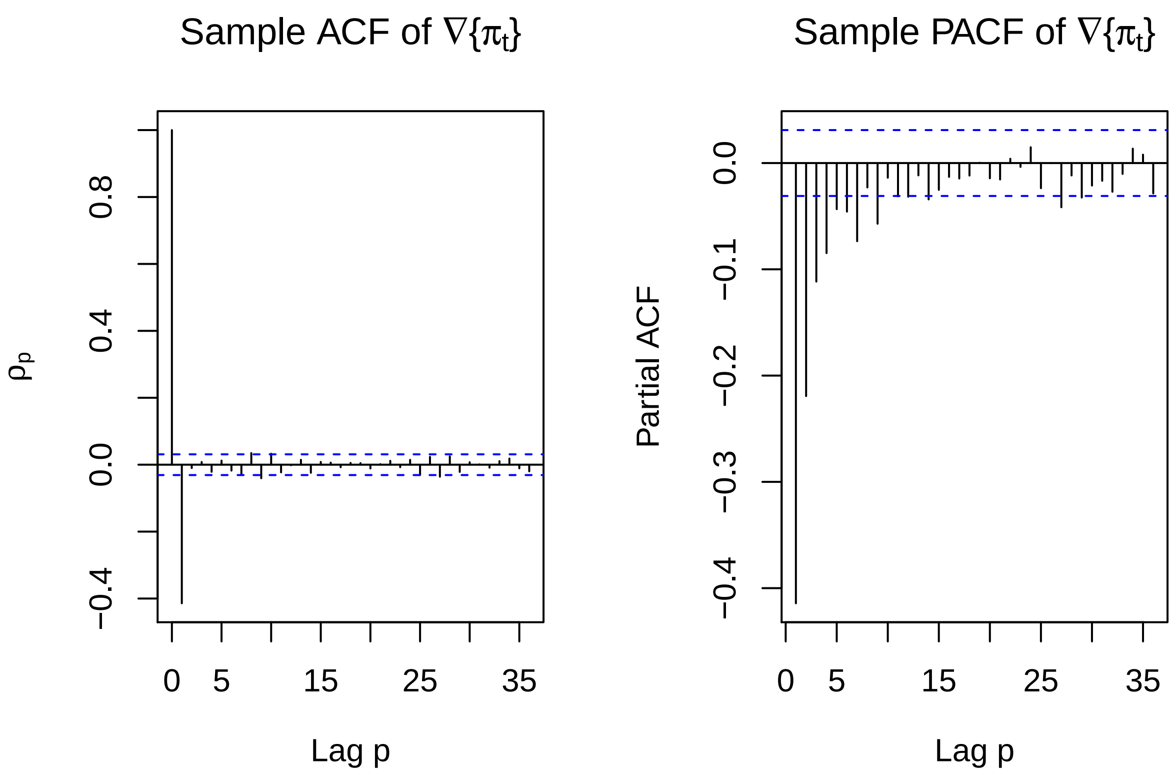


Figure 4: Sample ACF and PACF of $\nabla\{\pi_t\}$, which appear far more stationary.

### 6.3 Model construction and evaluation

A good model will yield residuals $\{e_t\}$ that look like zero mean white noise:

1. They are are uncorrelated. If there are correlations, the residuals contain information that may be used to estimate a better model.
2. They have zero mean. If they have a non-zero mean, then the model (and forecasts) are biased.
3. They have constant variance.

#### 6.3.1 IMA(1,1)

This is the simplest model of the three, and the simplest ARIMA model that seemed compatible with the data. When we fit the ARIMA$(0, 1, 1)$ model to the time series data, we get the following result.

```
##        ma1
## -0.5705635
```

To assess whether the model results in uncorrelated residuals, we inspect figure 5.

The histogram looks symmetric about 0 but the Q-Q suggests a lack of normality in the residuals. More telling, plots of the sample ACF and PACF of the residuals are evidence of high correlation. We reject this model.

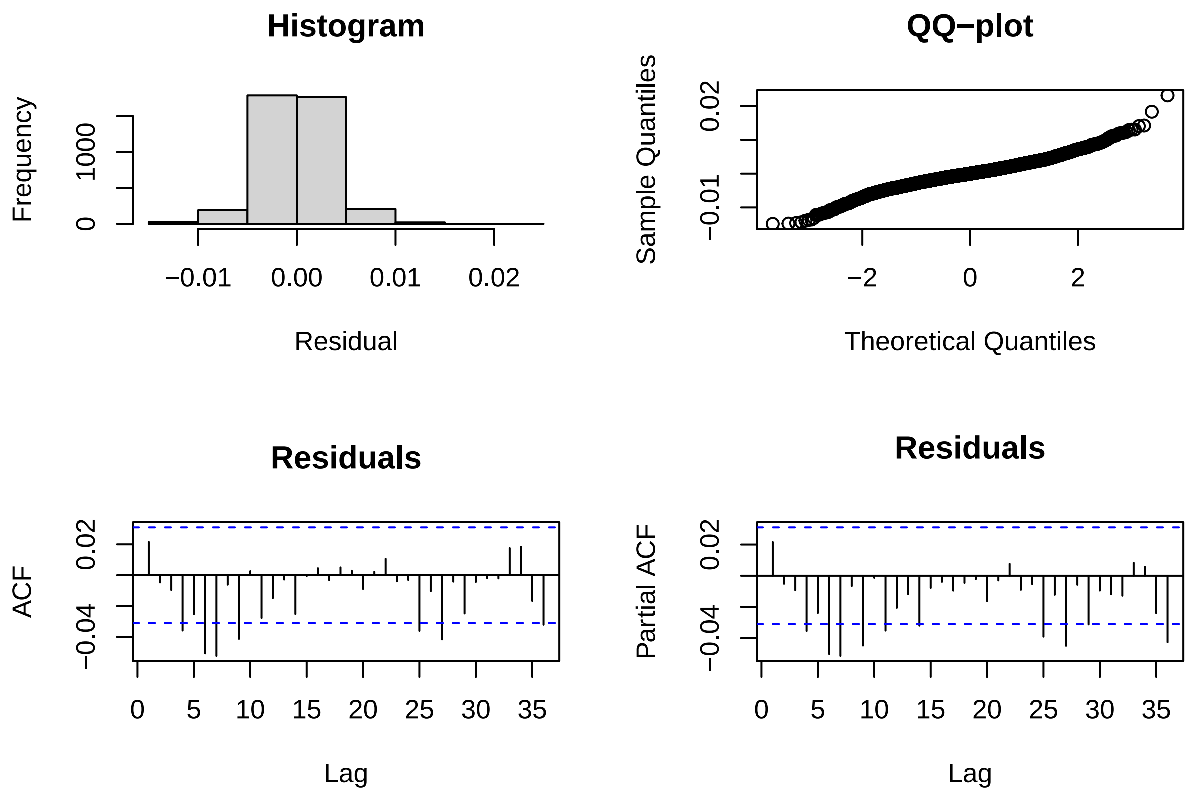


Figure 5: Plots to conduct stationary assessments of ARMA$(1,1)$.

#### 6.3.2 IMA(1,2)

When we fit ARIMA$(0,1,2)$ to the time series data, we get the following results.

```
##         ma1         ma2
## -0.55254743 -0.04124893
```

To assess whether the model results in uncorrelated residuals, we inspect the plots in figure 6.

The histogram looks symmetric about 0 but, again, the Q-Q suggests a lack of normality in the residuals. Note that this is not strictly required. More telling, plots of the sample ACF and PACF of the residuals are evidence of high correlation. We reject this model.

#### 6.3.3 ARIMA(1,1,2)

When we fit the ARIMA$(1,1,2)$ model to the time series data, we get the following results.

```
##        ar1        ma1        ma2
##  0.8839643 -1.4423574  0.4689439
```

To assess whether the model results in uncorrelated residuals, we inspect figure 7.

The histogram looks symmetric about 0 but, again, the Q-Q suggests a lack of normality in the residuals. However, the sample ACF and PACF seem reasonable. We perform the Ljung-Box hypothesis (lag = 10) test on the residuals of the model for a more objective measure of white noise.

```
## Box-Ljung test
```

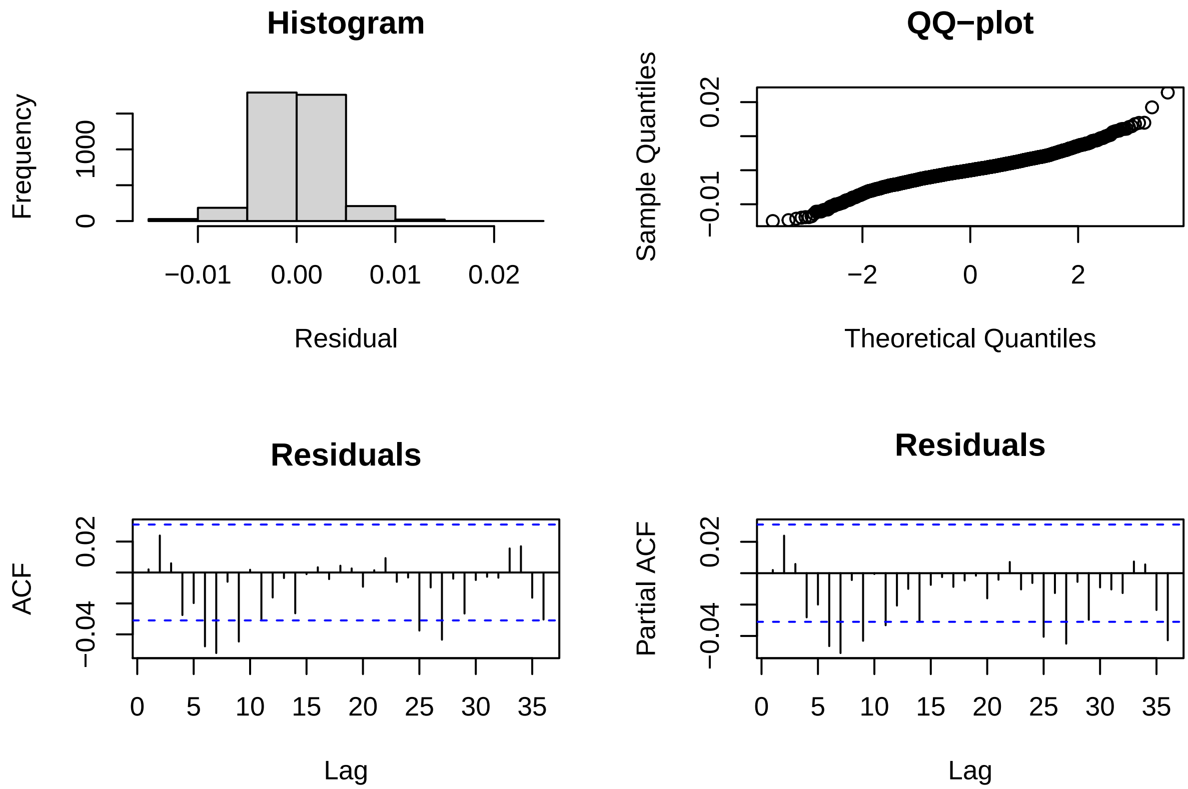


Figure 6: Plots to conduct stationary assessments for ARMA$(0,1,2)$.

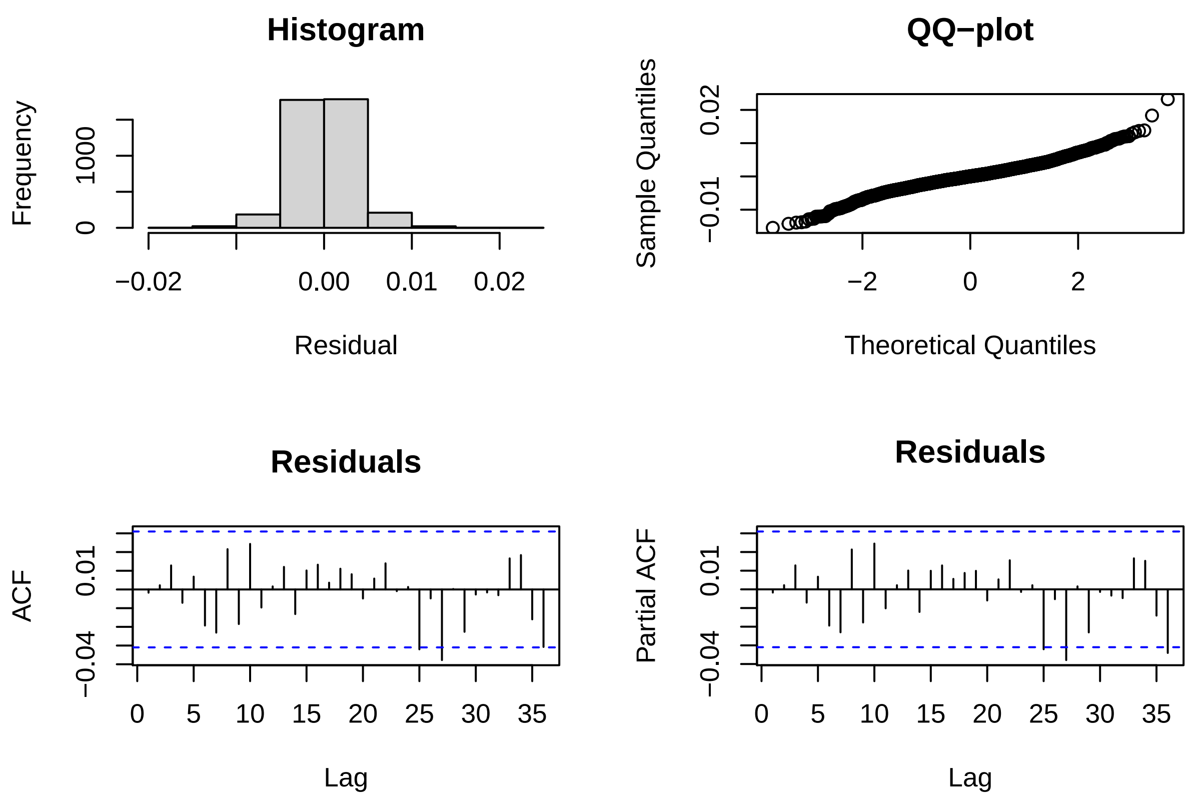


Figure 7: Plots to conduct stationary assessments of ARMA$(1,1,2)$.

```
## data: model.3$residuals
## X-squared = 10.333, df = 7, p-value = 0.1705
```

The null hypothesis is that the residuals for the model are white noise. The test reports a $p$-value of 0.171, which we consider to be strong evidence in support of the white noise hypothesis.

We choose this model. Since only one model seemed like a reasonable fit, measures like AIC were not needed.

What follows ia a summary of the chosen model.

```
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1     ma2
##       0.8840  -1.4424  0.4689
## s.e.  0.0276   0.0338  0.0266
##
## sigma^2 estimated as 1.088e-05:  log likelihood=17177.64
## AIC=-34347.27   AICc=-34347.26   BIC=-34322.1
```

We present it in the more familiar form given by

$$(1 - 0.884\,\mathrm{B})\nabla Y_t = (1 + 1.442\,\mathrm{B} - 0.469\,\mathrm{B}^2)e_t$$

or, equivalently,

$$(1 - 0.884\,\mathrm{B})\nabla Y_t = (1 - 0.273\,\mathrm{B})(1 + 1.715\,\mathrm{B})e_t$$

where $\{e_t\}$ is given by zero mean white noise,

$$e_t \sim \mathrm{WN}(\mu = 0, \sigma = 0.0033).$$

### 6.4 Forecasting

One of the primary goals of this time series analysis is forecasting, or predicting, the future accuracy of the adversary, i.e., $\hat{\pi}_{T+h|T}$. At $T = 5000$, we perform a forecast up to $h = 1000$ steps ahead, $\hat{\pi}_{T+h|T}$. See figure 8.

The held out *test* data remains within the 80% prediction interval for most of the time. All things considered, this seems like a reasonable forecast. However, due to what we believe may be a *model* misspecification, we think the prediction intervals are too wide and we have reason to believe that, in the long run, $\{\pi_t\}$ will decrease in variance and hover around some asymptotic limit. We explore this more in the next section.

### 6.5 Incorporating *a priori* information

We have *a priori* knowledge that we wish to incorporate into the model. Assuming the plaintext distribution $\{x_t\}$ is static and the algorithm that converts plaintext to ciphers is fixed, we observe the following:

1. The accuracy of the adversary, $\pi_t$, is a measure between 0 and 1.

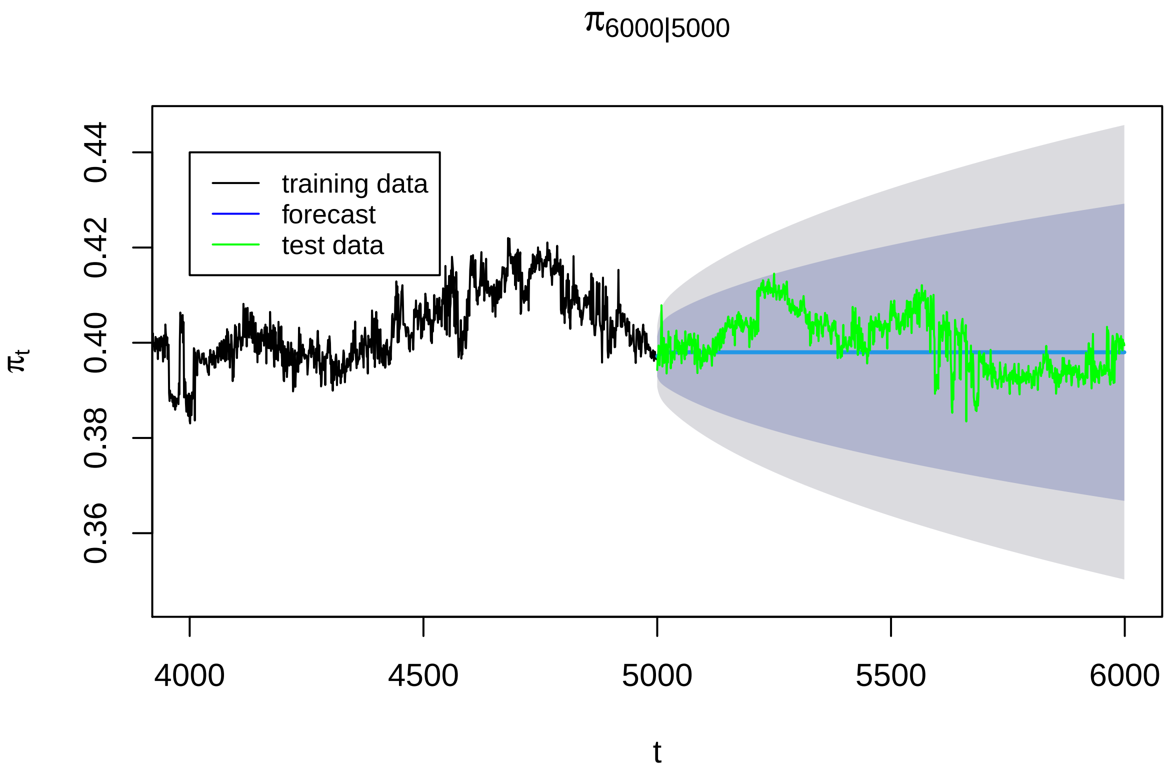


Figure 8: Forecasting the future with the training set and test sets superimposed.

2. Under ideal conditions, acquiring more knowledge by observing a larger sample is not expected to harm the adversary's accuracy $\pi_t$, in which case the expectation $\pi_t$ would be a monotonically increasing function that has an asymptotic limit $c \leq 1$.

Of course, at different points in time the adversary's accuracy may change due to, say, the presence of significant unaccounted covariates.

An ideal model for these axioms may be something like the Gompertz model or even a *scaled*, *relocated*, and *shifted* cumulative distribution function (cdf). However, for the sake of model simplicity, we assume a logarithmic form, which allows us to use linear regression. It does not have an asymptotic limit, but we hypothesize that it is a reasonable approximation for most finite time-horizons of interest.

Thus, we suppose the time series $\{\pi_t\}$ has the functional form

$$\pi_t = \beta_0 + \beta_1 \log t.$$

Instead of i.i.d. "errors" (deviations from the mean), we have reason to believe the errors are correlated. We choose to model these errors in the ARIMA family, such that $\{\Pi_t\}$ is a random process of the form

$$\Pi_t = \beta_0 + \beta_1 \log t + \eta_t$$

where

$$\eta_t \sim \text{ARIMA}(p, d, q).$$

Actually, this is not quite true, since according to [3], "The presence of lagged values [...] means that $\beta_1$ can only be interpreted conditional on the value of previous values of the response variable, which is hardly intuitive."

No matter, we press on and fit the model to the data, which yields the sought after ARIMA regression errors for $\{\eta_t\}$ and estimates for the parameters.

```
## Regression with ARIMA(1,1,2) errors
##
## Coefficients:
##          ar1      ma1     ma2    xreg
##       0.8521  -1.4012  0.4355  0.0292
## s.e.  0.0234   0.0277  0.0211  0.0202
##
## sigma^2 estimated as 6.828e-06:  log likelihood=45279.87
## AIC=-90549.73   AICc=-90549.73   BIC=-90513.68
```

We have used the same model as before, except with the dynamic regression on the logarithm of time $t$. It turns out that, for the dynamic regression, ARIMA$(2,1,2)$ does better and enjoys tighter prediction intervals as well, but it was not significantly better.

To assess whether the model results in uncorrelated residuals, we insepect figure 9.

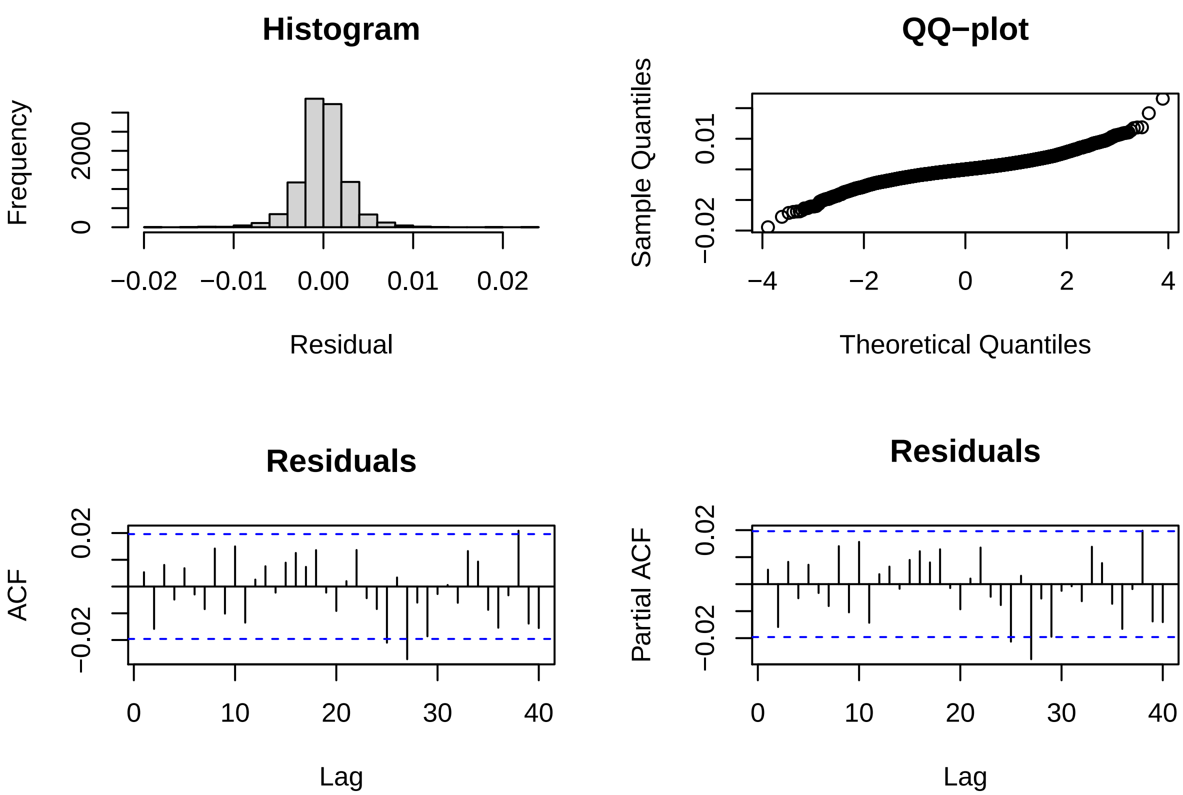


Figure 9: Plots to conduct stationary assessments

The histogram looks symmetric about 0 but, again, the Q-Q suggests a lack of normality in the residuals. However, the sample ACF and PACF seem reasonable. We perform the Ljung-Box hypothesis (lag = 10) test on the residuals of the model for a more objective measure of white noise.

```
## Box-Ljung test
## data:  reg_model$residuals
## X-squared = 10.279, df = 6, p-value = 0.1134
```

At the 5% significance level, this model barely passes. We see that

$$\hat{\Pi}_t = 0.029 \log t + \eta_t$$

where

$$\eta_t \sim \text{ARIMA}(5, 1, 1)$$

with the above specified estimated coefficients and

$$e_t \sim \text{WN}(\mu = 0, \sigma = 0.0026).$$

We show a time series plot of the model with both the training set (in black) and the test set (in green) superimposed onto it in figure 10.

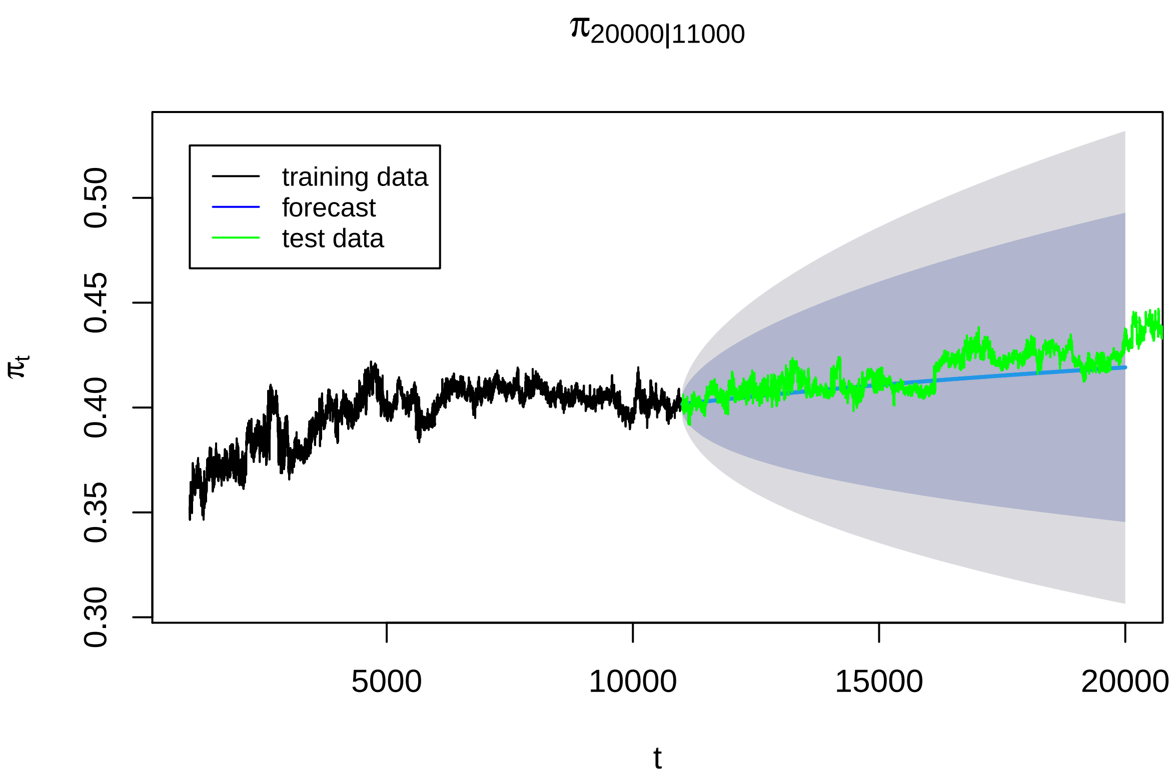


Figure 10: Forecast distribution of more appropriate theoretical model of the time series.

We have forecasted much further into the future. The forecast seems reasonable, as it follows the subtle positive non-linear trend.

When we compare it to the previous ARIMA model, we see that the subtle positive trend is not captured by the model. We can address this potential shortcoming by forcing the ARIMA model to include the drift term. When we do this, we get the following results.

```
##           ar1           ma1           ma2         drift
##  8.472755e-01 -1.396547e+00  4.324823e-01  4.695287e-06
```

We see that the drift term is a positive value near 0, but over a sufficiently long period of time it adds up, as demonstrated by the figure 11.

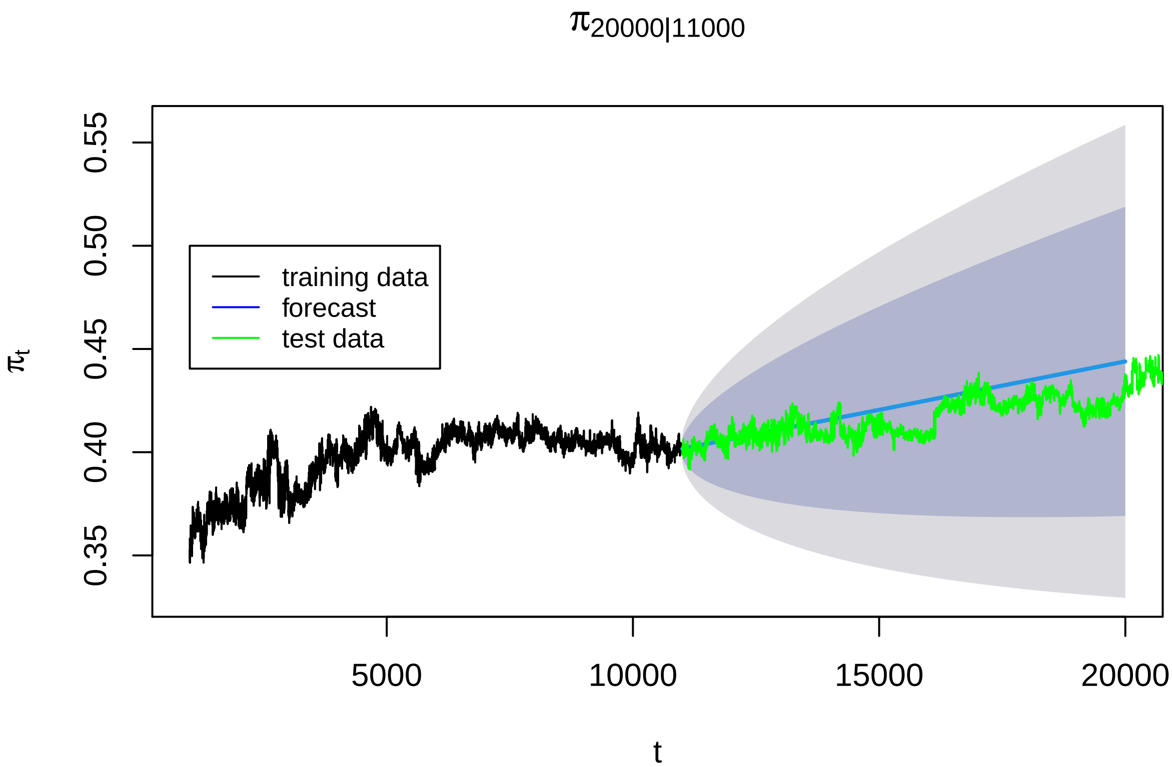


Figure 11: Forecast dsitribution of the ARIMA model with drifting.

The ARIMA model with drift adds a linear element to the autoregression, which theoretically is not appropriate.

## 7 Future work: dynamic regression on co-variates

In our time series analysis, the forecasting model only used lagged values of the confidentiality measure to forecast future values and we made no attempt to discover any other co-variates. Therefore, it extrapolated trends but ignored any other information such as *side-channel* information that may help or hinder the adversary's efforts to decode the ciphers.

At a time $t'$, the adversary may learn something about the system other than observing the time series of ciphers, $\{C_t\}$. This information may be incorporated into the time series model through an autoregression that has predictor variables other than just lagged components of the measure on the adversary's accuracy, $\{\pi_t\}$. The estimated paramters of the dynamic autoregressive model may also be used to explain the effect such predictor variables have on confidentiality.

A potentially interesting model is given by the data

$$(t, \pi_t, I_t)$$

where $t$ denotes time index, $\pi_t$ denotes the adversary's accuracy at time $t$, and $I_t$ denotes the *information* measure of the $t$-th observation, defined as

$$I_t = \log_2 \frac{1}{\Pr(g(c_{t'}))}.$$

Observe that lagged components of $I_t$ may be used to make the regression a function of entropy $H_t$ as well.

When the entropy is reduced or an informative observation comes in, this may have a larger impact on the time series $\{\pi_t\}$ and ideally we would incorporate this effect into the model.

The information gain does not necessarily need to be related to any of the time series previously mentioned, either. For instance, suppose the adversary, through side-channel information, acquires the knowledge that a certain cipher $c'$ maps to some smaller subset $\mathbb{W} \subset \mathbb{X}$. This also may be modeled as an information gain or entropy reduction, since the distribution of ciphers $\{c_t\}$ has less entropy given this information.

# 8 Conclusion

The statician George Box once wrote, "All models are wrong, some are useful." If we include *drifting* in the ARIMA model, it eventually predicts impossible futures. More to the point, it is not a good match for the theoretical model, as its bias is a function of time $t$.

The logarithmic model performs better in this regard, as it takes an inordinately long time (1,531,520,000,000,000 steps) to reach impossible values, although the prediction interval obtains it much more quickly. In addition, it more closely matches the features of the theoretical underlying distribution.

That said, there is still a lot to be said of the ARIMA$()1,1,2)$ model, given its simplicity. The adversary takes a very long time before it starts to seem like the simple model may be negatively biased.

Recall that if we specify that $\pi^* = 0.44$ is the minimum confidentiality we wish to maintain, a reasonable policy may be to use the latest observation to forecast the future to estimate where $\pi_t = \pi^*$, which is given by

$$\hat{T}^* = \arg\min_T \pi_{T \mid 11000} > \pi^*.$$

Inspecting figure 12, we see that $\hat{T}^* \approx 20000$. That is, to maintain $\pi^* > 0.44$, a password reset should occur before $T^* \approx 20000$.

Interestingly, the prediction intervals are far less forgiving and if we used those as a pessimistic estimator of $T^*$, we would nearly instantaneouls need to do a password reset.

To be a useful measure, it would seem that the uncertainty should be lower. Possibly, as we discuss in section 7, we could incorporate other covariates that help reduce the uncertainty. Or, perhaps we need to impose a more realistic dynamic trend. After all, the logarithm is not a particularly good fit for the data either, it is simply potentially better than the other alternatives.

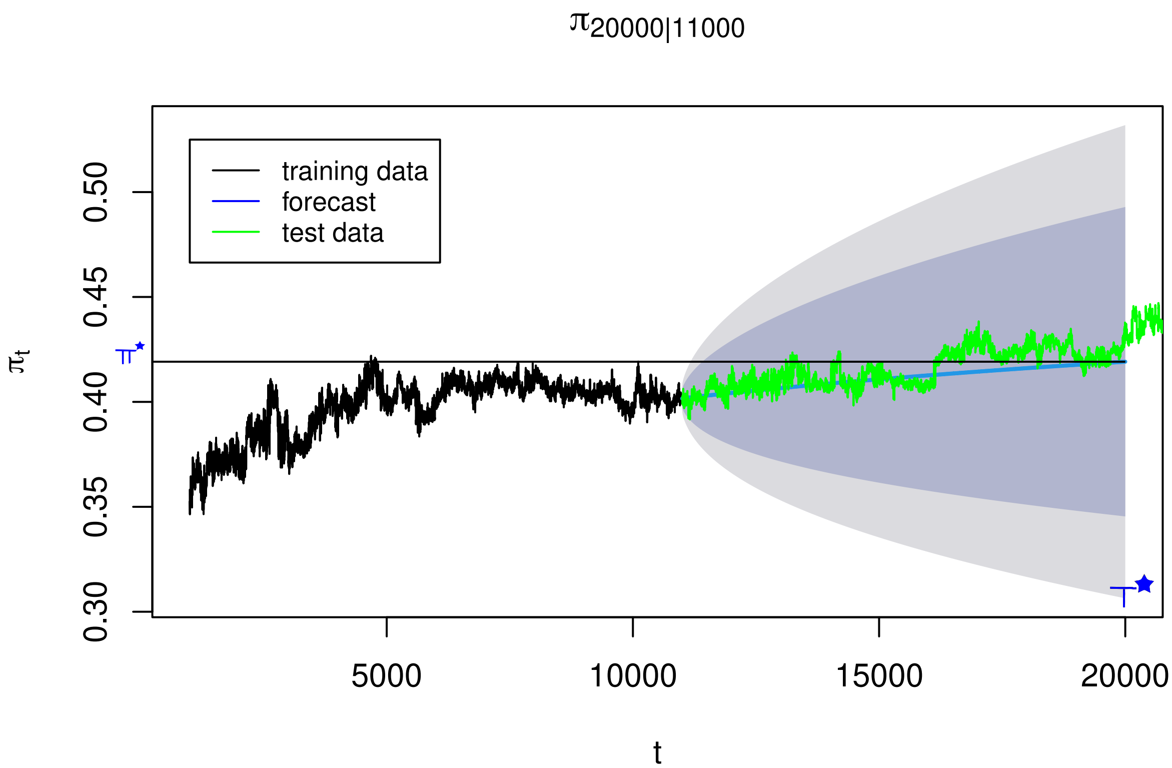


Figure 12: $\pi^*$ vs $T^*$.

## References


[1] Bias–variance tradeoff, April 2021. Page Version ID: 1018809373. URL: https://en.wikipedia.org/w/index.php?title=Bias_variance_tradeoff&oldid=1018809373.

[2] Dickey–Fuller test, April 2021. Page Version ID: 1019701572. URL: https://en.wikipedia.org/w/index.php?title=Dickey_Fuller_test&oldid=1019701572.

[3] Hundman Rob. The ARIMAX model muddle, October 2010. URL: https://robjhyndman.com/hyndsight/arimax/.